

# EPS 2.0

---

## Quick User Guide

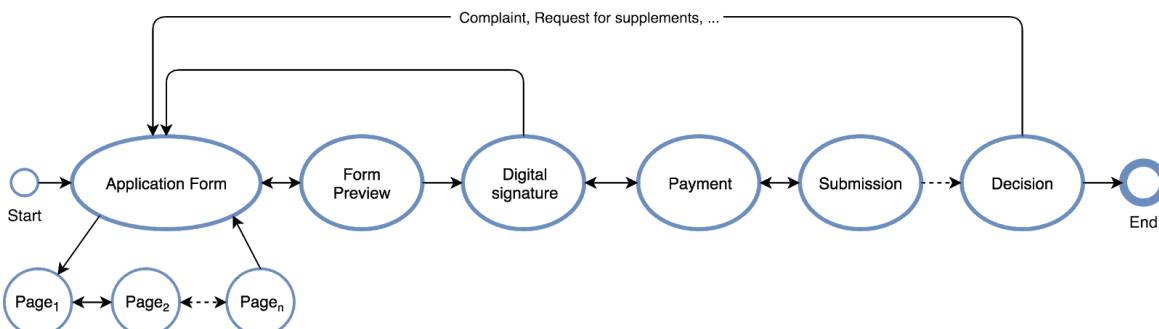
Tip dokumenta	Reference documentation
Number	EPS2.0_QUG_20190313_v1.1
Author	Medius d.o.o.
Recipient	g. Boštjan Tovornik Ministrstvo za javno upravo Tržaška 21 1000 Ljubljana
Date	13.3.2019
Author	Petar Brajak, Viktor Brajak
Version	1.1

<b>1. Introduction</b>	<b>2</b>
<b>2. Jedro-admin</b>	<b>8</b>
2.1. Creating procedures	9
2.1.1. Process definition	12
2.1.2. Procedure actions customization	13
<b>3. Jedro-front and Jedro-front-frame</b>	<b>19</b>
<b>4. Application Form Editor</b>	<b>23</b>
4.1. XML presentation	26
4.2. Previews	27
4.3. Graphical components	28
4.3.1. Layouts	28
4.3.1.1. Page	28
4.3.1.2. Group	31
4.3.1.3. Repeat	32
4.3.1.4. Tabsheet	34
4.3.2. Input components	35
4.3.2.1. Label	35
4.3.2.2. Text	36
4.3.2.3. 4.3.2.3.Combo	37
4.3.2.4. Options	38
4.3.2.5. Date Components	39
4.3.2.6. Table	40
4.3.2.7. Table tree	40
4.3.2.8. Action	42
4.3.2.9. Search	42
4.3.2.10. Upload	43
4.3.3. Rules	44
4.3.4. Validations	45
4.3.5. Data Sources	46
4.3.5.1. Mapping data	48
4.3.6.1. Repeatable data gathering	50
4.3.7. Groovy Script	51
<b>5. Software as a Service</b>	<b>52</b>

5.1. Technology	54
<b>6. MediusFormsDemo</b>	<b>54</b>
<b>7. Video demos</b>	<b>59</b>

# 1. Introduction

Information System JEP - Jedro Elektronskih Postopkov or EPS - Electronic Procedures System in English is based on the concept of modeling business processes (BPM), called EPS procedures. Each EPS procedure consists of a combination of process state nodes (actions) and transitions between the nodes, orchestrated by the custom developed Process State Engine. Picture 1.1 demonstrates a typical process definition that can be used to create different EPS procedures. Process State Engine instantiates the procedure at runtime based on its process definition. The procedure is visualized as an ordered set of traversable process nodes, each node represented by one or more HTML pages and the associate application logic.



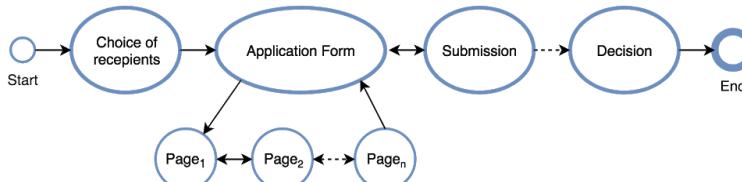
Picture 1.1 - Generic process definition for a typical application form submission

A process definition as shown in picture 1.1 has been used for creating numerous procedures and application forms at Ministry of Public Administration, Republic Slovenia. This hypothetical process definition defines the following set of process state nodes/actions:

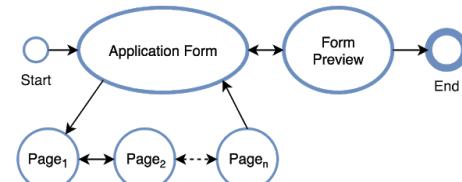
- Application Form - an ordered set of pages designed by the application form editor as described in chapters 2 and 4. Administrators design application forms using this WYSIWYG canvas based editor that contains of various graphical components for data entry fields, mapping of entry fields and data gathered from the external SOAP web services, data validation, rules, form logic, etc.
- Form preview - HTML render of the application form and data entered by the applicant during the form filling.

- Digital signature - integration with the central digital signature system SI-CES. IS EPS digitally signs the PDF representation of the application form and the corresponding metadata (XML form data representation).
- Payment - integration with the official Slovene Payment Gateway.
- Submission - choice of different submission types:
  - submission to the official SI-CEV submission service,
  - submission to the recipient's SOAP web service,
  - submission to the internal EPS inbox service which offers an open API for recipient's applications to get application forms.
- Decision node - a programming logic that manages the communication with the competent organization and maintains the procedure states for complaints, requests for supplements, cancellations, etc.

Process definitions can contain any combination of predefined (preprogrammed) process nodes and transitions. Pictures 1.2 and 1.3 demonstrate two different types of process definitions. For example, process definition in picture 1.2 is used for the procedures that do not require the digital signature and payment, but require the action of choosing a recipient. On the other hand, a process definition in picture 1.3 is used for procedures that gather data from the official registers via SOAP web services and render that data on the application form.

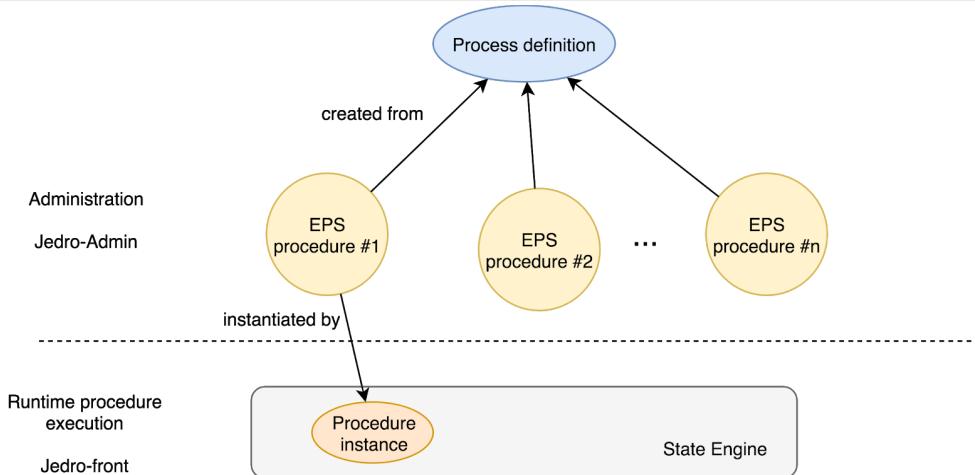


Picture 1.2 - Choice node



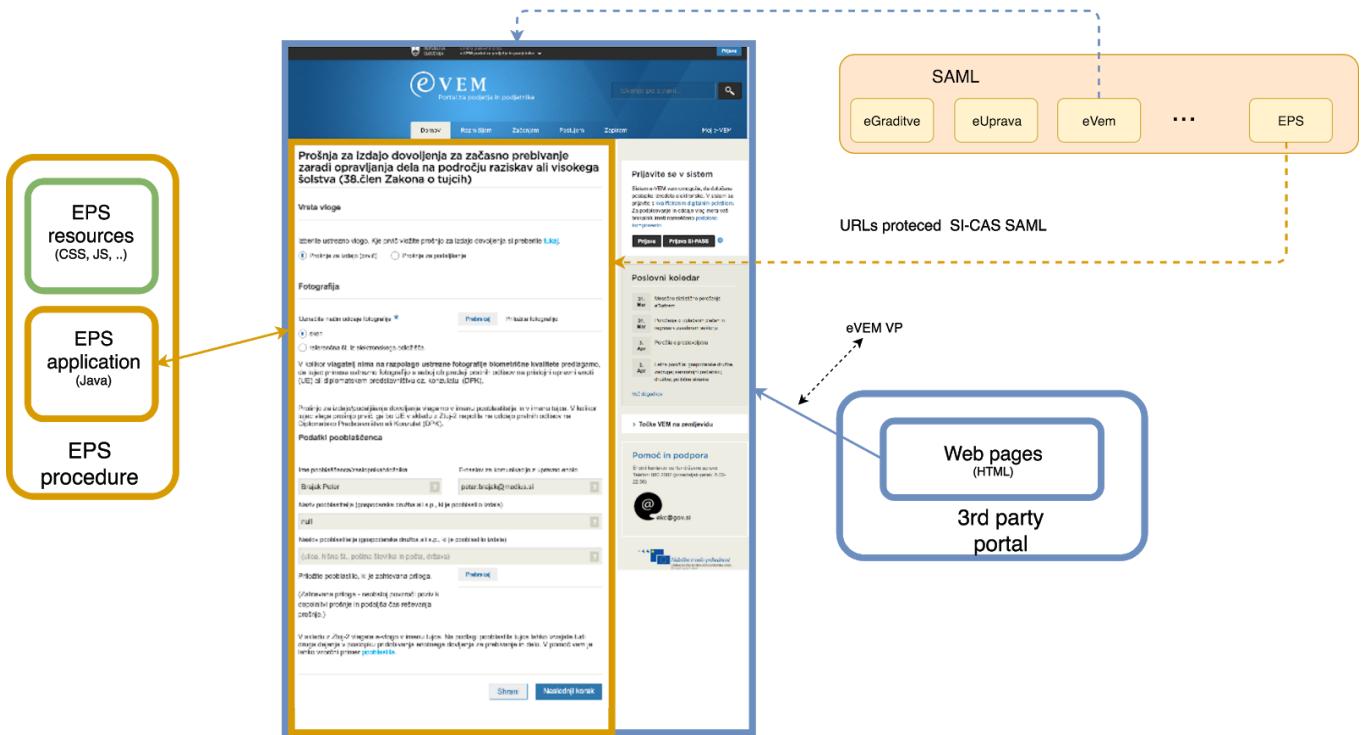
Picture 1.3 - Data gathering form

The runtime execution of the procedures is orchestrated by the custom developed Process State Engine which renders the procedure as a set of traversable HTML pages. The relationships between the process definition, procedures and the procedure instance is shown in picture 1.5. Different procedures can be created from the same process definition.



Picture 1.5 - Process definition, procedure, process instance

When a procedure is instantiated by the Process State Engine the resulting HTML pages can be embedded into any 3rd party portal or a HTML page using a technique called CORS (Cross Origin Resource Sharing). Picture 1.6 shows an example of a typical 3rd party portal (blue frame) that embeds the running of EPS procedure by using the CORS technique (yellow frame).



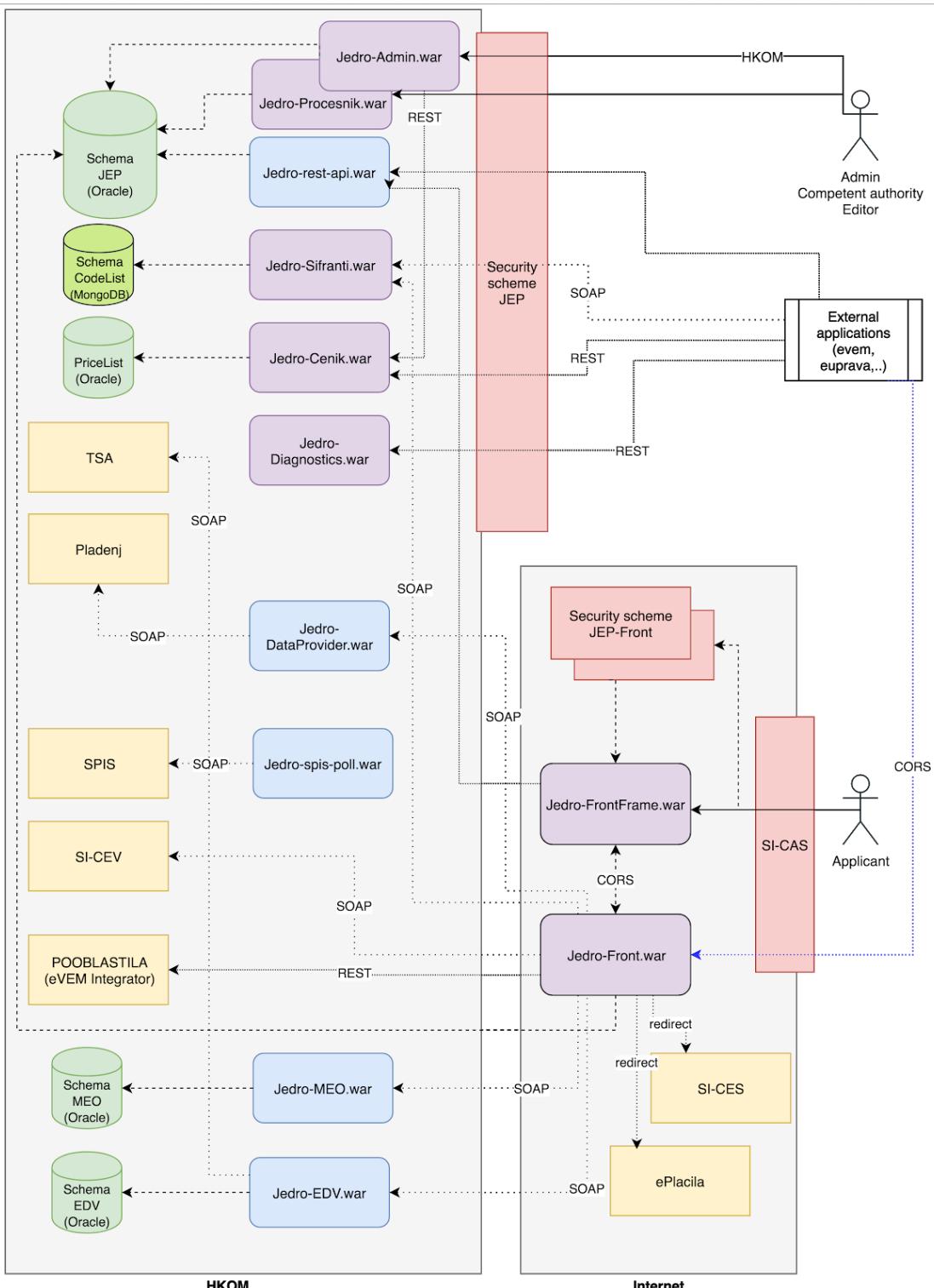
Picture 1.6 - CORS (Cross Origin Resource Sharing)

---

This user guide is focused on how to manage procedures, process definitions and the application forms only. It does not describe the CORS integration techniques, the security issues related to CORS and especially, it does not describe the API and the library of JavaScript functions available for the 3rd party developers so that they can easily and seamlessly embed EPS procedures into their HTML pages or portals. The CORS documentation can be located at <https://mju-jep.test.mediush.si/api/docs/index.html> (Medius) or <https://jep-test.sigov.si/api/docs/index.html> (MJU).

However, this documentation gives an overview of the architecture concepts, the SaaS (Software as a Service) nature of the whole information system and the relationships between the most important system components Jedro-admin, Jedro-front and Jedro-front-frame. Word "jedro" means core in the Slovenian language. Therefore, IS EPS is considered the core system for creating and submitting application forms for different organizations and can be used for multi-institutional, nationwide and even cross-border purposes.

IS EPS is deployed as SaaS (Software As A Service) cloud based information system. It consists of 12 container based microservices (build by Docker), each one separately deployed in the private Slovene State Cloud, orchestrated by the Kubernetes open-source orchestration system. Each microservice defines its own policy for upscaling, downscaling, rolling updates, load balancing, etc. Picture 1.5 shows the whole architecture and all microservices.

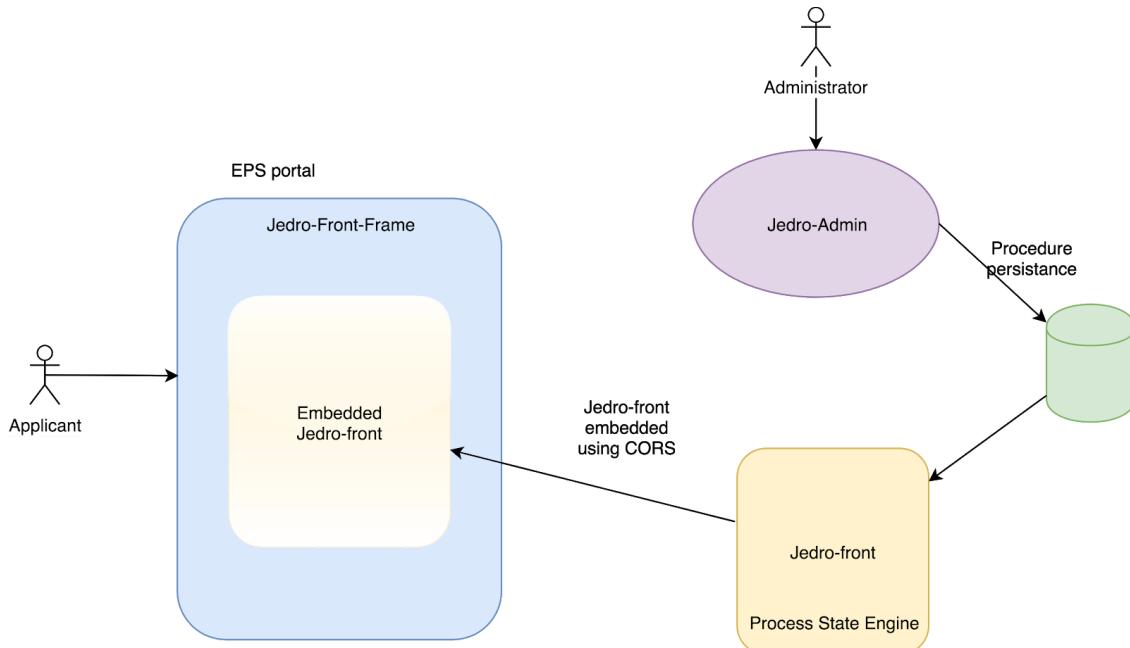


Picture 1.5 - Cloud based architecture of IS EPS

In this documentation we focus only on 3 microservices which are responsible for managing procedures, process definitions, application form. These 3 microservices are:

- **Jedro-admin** - administration microservice with tools and editors to create process definitions, application forms, procedures, etc.
- **Jedro-front** - microservice that renders EPS procedures created by Jedro-admin. Jedro-front runs the Process State Engine that renders the procedure. The resulting procedure's HTML pages are embedded into the portal called Jedro-front-frame using the CORS techniques. Jedro-front can be embedded into any HTML page that supports HTML iframe tag and JavaScript. Jedro-front provides API and JavaScript library for 3rd party developers for seamless embedding of Jedro-front into their HTML environment.
- **Jedro-front-frame** - simple EPS portal for rendering procedures running in Jedro-front. It uses Jedro-front JavaScript API and library to embed the procedures. In general, any other portal or HTML pages can be used for rendering Jedro-front procedures instead of Jedro-front-frame,

The following picture demonstrates the relationship between Jedro-admin, Jedro-front and Jedro-front-frame.



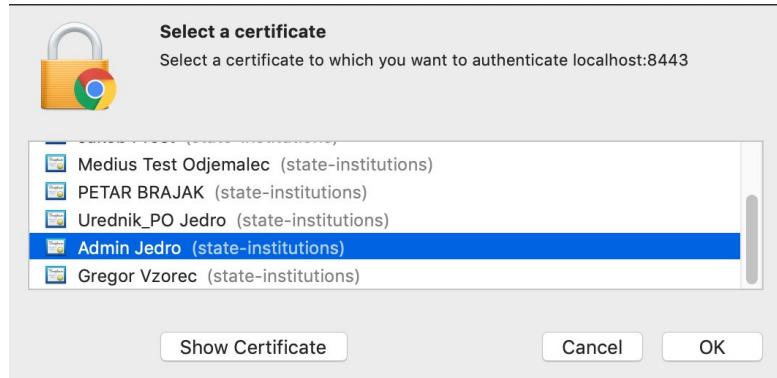
Picture 1.6 - Three IS EPS microservices used for creating and rendering procedures

## 2. Jedro-admin

Jedro-admin is a SaaS type application used to create EPS process definitions, procedures and application forms. It is used also for other administration tasks such as monitoring, setting up the environment, data analysis, health checking, etc.

Administrators can login into the system only if they are authenticated and authorised by the Central Security Agency of Ministry of Public Administration. Digital certificates are used for authentication. Please request for the admin rights at

<https://vs-sola.gov.si/VS.web/prosnjaDodelPrav> or contact [jep-admin@mju.gov.si](mailto:jep-admin@mju.gov.si). For example, the certificate “Admin Jedro”, as shown in picture 2.1 has the administration privileges and this role can manage all procedures. In general, the administration rights for Jedro-admin are defined per organisation, so only the responsible organization can manage its own procedures and application forms.



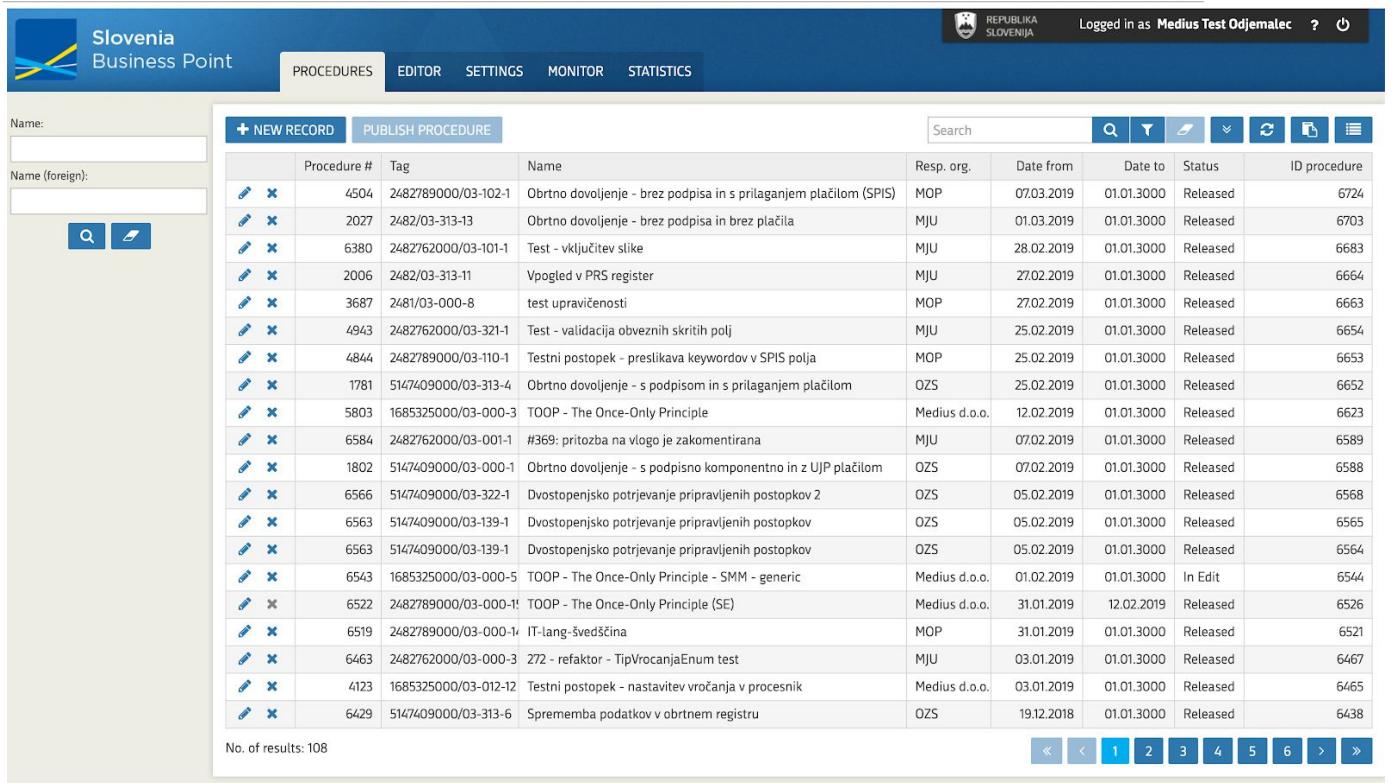
Picture 2.1 - Required certificate at login

Two testing EPS information system environments are available:

- at Medius (development): <https://mju-jep.test.mediust.si/admin/?lang=en>
- at Ministry of Public Administration (testing): <https://jep-test.gov.si/admin/?lang=en>

A suffix [?lang=en](#) in the URL is optional and is used for changing the application language.

After a successful login, the following landing page appears. It displays all procedures the administrator is responsible for.



The screenshot shows the Medius Admin landing page. At the top, there's a header with the Slovenia Business Point logo, navigation tabs for PROCEDURES, EDITOR, SETTINGS, MONITOR, and STATISTICS, and a user logged in as 'Medius Test Odjemalec'. Below the header is a search bar and a toolbar with various icons for filtering, ordering, and exporting data. The main content area is a table listing 30 procedures. Each row contains columns for Procedure #, Tag, Name, Resp. org., Date from, Date to, Status, and ID procedure. Buttons for 'Edit' (pencil) and 'Delete' (trash) are present in each row. At the bottom of the table, it says 'No. of results: 108' and shows a page navigation bar with buttons for 1 through 6.

	Procedure #	Tag	Name	Resp. org.	Date from	Date to	Status	ID procedure
	4504	2482789000/03-102-1	Obrtno dovoljenje - brez podpisa in s prilaganjem plačilom (SPIS)	MOP	07.03.2019	01.01.3000	Released	6724
	2027	2482/03-313-13	Obrtno dovoljenje - brez podpisa in brez plačila	MJU	01.03.2019	01.01.3000	Released	6703
	6380	2482762000/03-101-1	Test - vključitev slike	MJU	28.02.2019	01.01.3000	Released	6683
	2006	2482/03-313-11	Vpogled v PRS register	MJU	27.02.2019	01.01.3000	Released	6664
	3687	2481/03-000-8	test upravičenosti	MOP	27.02.2019	01.01.3000	Released	6663
	4943	2482762000/03-321-1	Test - validacija obveznih skritih polj	MJU	25.02.2019	01.01.3000	Released	6654
	4844	2482789000/03-110-1	Testni postopek - preslikava keywordov v SPIS polja	MOP	25.02.2019	01.01.3000	Released	6653
	1781	5147409000/03-313-4	Obrtno dovoljenje - s podpisom in s prilaganjem plačilom	OZS	25.02.2019	01.01.3000	Released	6652
	5803	1685325000/03-000-3	TOOP - The Once-Only Principle	Medius d.o.o.	12.02.2019	01.01.3000	Released	6623
	6584	2482762000/03-001-1	#369: pritožba na vlogo je zakomentirana	MJU	07.02.2019	01.01.3000	Released	6589
	1802	5147409000/03-300-1	Obrtno dovoljenje - s podpisno komponentno in z UJP plačilom	OZS	07.02.2019	01.01.3000	Released	6588
	6566	5147409000/03-322-1	Dvostopenjsko potrjevanje pripravljenih postopkov 2	OZS	05.02.2019	01.01.3000	Released	6568
	6563	5147409000/03-139-1	Dvostopenjsko potrjevanje pripravljenih postopkov	OZS	05.02.2019	01.01.3000	Released	6565
	6563	5147409000/03-139-1	Dvostopenjsko potrjevanje pripravljenih postopkov	OZS	05.02.2019	01.01.3000	Released	6564
	6543	1685325000/03-000-5	TOOP - The Once-Only Principle - SMM - generic	Medius d.o.o.	01.02.2019	01.01.3000	In Edit	6544
	6522	2482789000/03-000-1!	TOOP - The Once-Only Principle (SE)	Medius d.o.o.	31.01.2019	12.02.2019	Released	6526
	6519	2482789000/03-000-1!	IT-lang-švedščina	MOP	31.01.2019	01.01.3000	Released	6521
	6463	2482762000/03-000-3	272 - refaktor - TipVrocjanjaEnum test	MJU	03.01.2019	01.01.3000	Released	6467
	4123	1685325000/03-012-12	Testni postopek - nastavitev vročanja v procesnik	Medius d.o.o.	03.01.2019	01.01.3000	Released	6465
	6429	5147409000/03-313-6	Sprememba podatkov v obrtnem registru	OZS	19.12.2018	01.01.3000	Released	6438

Picture 2.2 - Admin landing page

Buttons in the upper right corner are used for filtering, ordering and exporting procedure data displayed in the table view. Button “Pencil” is used to edit selected procedure’s metadata.

This documentation can be downloaded by clicking the “?” button in the upper right corner of the landing page.



## 2.1. Creating procedures

Button is used to create a new procedure. It opens a dialog that is used to enter relevant procedure metadata, determine the process definition, create an application form, determine the payment options for the application form, choose the payment method, submission type, etc.

New record + ×

Name: *	Moj novi testni postopek								
Foreign language: *	Angleščina (en)								
Name (foreign): *	My new test procedure								
Competent organization: *	2482762000 - MJU	Prepared by the * organization:	2482762000 - MJU						
Class. category:	0 - DRŽAVNA IN DRUŽBENA UREDITEV								
Class. subcategory:	00 - USTAVNA IN DRŽAVNA UREDITEV								
Classification number: *	000 - Splošno o ustavi in državni ureditvi								
Description:	<div style="border: 1px solid #ccc; padding: 5px; height: 100px;"></div>								
Roles:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: right;">☰</td> </tr> <tr> <td colspan="2" style="text-align: right;">×</td> </tr> </tbody> </table>			Name	Description	☰		×	
Name	Description								
☰									
×									
More info (URL with * http://):	www.medijs.si								
More info (foreign) (URL with http://):	www.medijs.si								
Date from:	<input type="button" value="Calendar"/>	Immediately active: <input checked="" type="checkbox"/>							
Process definition: *	<div style="border: 1px solid #ccc; padding: 5px; height: 100px;"></div>								
Type of procedure:	Classic								
Process actions:	The name of the step in the process	The order ▲	Feature						
<input type="button" value="OK"/> <input type="button" value="CANCEL"/>									

Picture 2.3 - Procedure metadata

The following is the list of procedure's required metadata fields:

- Name - procedure name in the first language (Slovene). Any text.

- Foreign language - combo-box of supported languages: English, German, Swedish, Croatian and 2 official national minority languages (Hungarian, Italian). *Choose English - Angleščina (en)*.
- Name (foreign) - procedure name in the second language. *Any text*.
- Competent organization - organization which is responsible for the procedure. *Choose MJU*.
- Prepared by organization - organization which prepares the procedure. *Choose MJU*.
- Procedure classification - used to generate proper and unique (nationwide) procedure tag/code. *Choose any combination*.
- Description - optional.
- Roles - used for the special kind of procedures used by internal staff at Ministry of Public Administration. *Do not use it*.
- More info - URL where additional information about the procedure can be found. *Use any valid www*.
- More info (foreign) - URL about the procedure in the second language. *Use any valid www*.
- Date from - starting date of the procedure. *Use "Immediately active"*.
- Process definition - choose one from the available process definitions. Note that different procedures can be created from the same process definition. Choose "*My new process definition*", or create a new one in the main menu SETTINGS → PROCESS DEFINITION. If you choose "*My new process definition*" the following simple process definition will be used for the procedure:

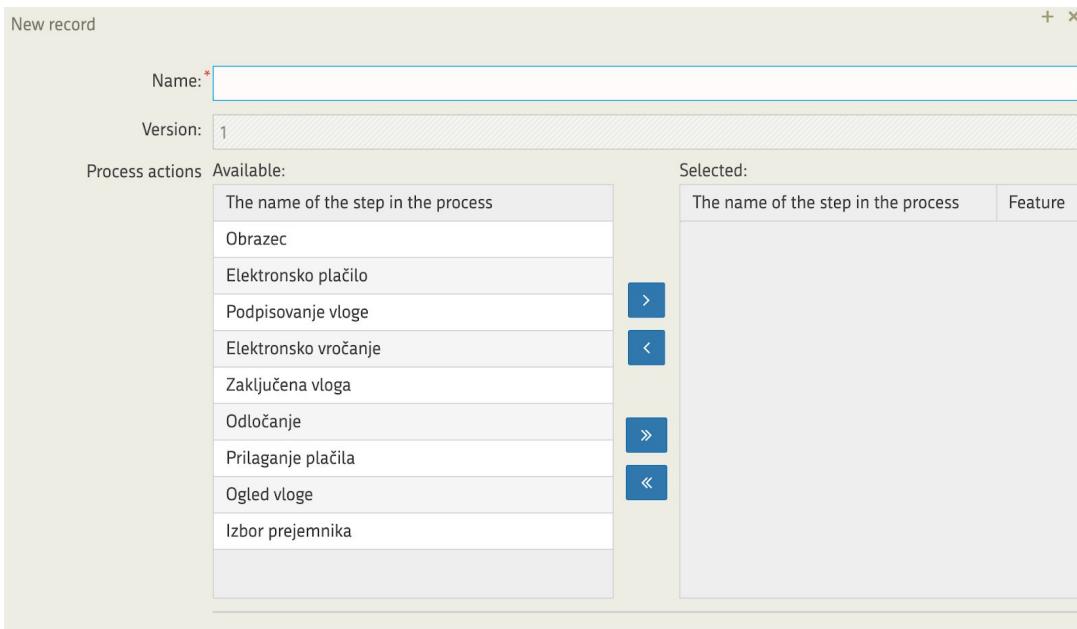
- Obrazec - (eng. Application Form),
- Ogled vloge - (eng. Application Form Preview),
- Elektronsko vročanje - (eng. Electronic Submission),
- Odločanje - (eng. Decision),
- Zaključevanje vloge - (eng. End of Process).

Process definition: *	My new process definition (1)	▼	
Type of procedure:	Classic	▼	
Process actions:	The name of the step in the process	The order ^	Feature
	Obrazec	1	
	Ogled vloge	2	
	Elektronsko vročanje	3	
	Odločanje	4	
	Zaključena vloga	5	

Picture 2.4 - Process definition selection

## 2.1.1. Process definition

In general, process definitions should be created only by the experienced EPS administrators. Therefore, it is not advised to create a process definition just for the sake of testing. For purpose testing, it is more appropriate to use one of the available process definitions, for example “*My new process definition*”. The following mask appears when a new process definition is to be created in the main menu SETTINGS → PROCESS DEFINITION.



Available:	
The name of the step in the process	
Obrazec	
Elektronsko plačilo	
Podpisovanje vloge	
Elektronsko vročanje	
Zaključena vloga	
Odločanje	
Prilaganje plačila	
Ogled vloge	
Izbor prejemnika	

Selected:	
The name of the step in the process	Feature

Picture 2.5 - New process definition creation

Available process action of the process definition are in the left part of the subset-selector component. At the time of writing this document the following actions are available: Obrazec (*Application Form*), Elektronsko plačilo (*Electronic Payment*), Podpisovanje vloge (*Digital Signature*), Elektronsko vročanje (*Electronic Submission*), Zaključevanje vloge (*Procedure End*), Odločanje (*Decision*), Prilaganje plačila (*Payment with Receipt*), Ogled vloge (*Preview*), Izbor prejemnika (*Choice of recipients*). Administrators can drag and drop process actions from the left to the right part of the subset selector, and vice versa. Button “*Feature*” is used to define transitions between the process states.

Let's look the process definition “*My new process definition*” in picture 2.6. The process definition contains 5 out of 9 process actions and two transitions (from “*Preview*” to “*Application form*” and from “*Electronic submission*” to “*Preview*”).

Edit + X

Name:	My new process definition																				
Version:	2																				
Process actions	Available:	Selected:																			
	<table border="1"> <tr><td>The name of the step in the process</td></tr> <tr><td>Elektronsko plačilo</td></tr> <tr><td>Podpisovanje vloge</td></tr> <tr><td>Prilaganje plačila</td></tr> <tr><td>Izbor prejemnika</td></tr> <tr><td> </td></tr> </table>	The name of the step in the process	Elektronsko plačilo	Podpisovanje vloge	Prilaganje plačila	Izbor prejemnika		<table border="1"> <thead> <tr><th>The name of the step in the process</th><th>Feature</th></tr> </thead> <tbody> <tr><td>Obrazec</td><td>+ <span style="color: blue;">+</span></td></tr> <tr><td>Ogled vloge</td><td>+ <span style="color: blue;">+</span></td></tr> <tr><td>Elektronsko vročanje</td><td>+ <span style="color: blue;">+</span></td></tr> <tr><td>Odločanje</td><td>+ <span style="color: blue;">+</span></td></tr> <tr><td>Zaključena vloga</td><td>+ <span style="color: blue;">+</span></td></tr> </tbody> </table>	The name of the step in the process	Feature	Obrazec	+ <span style="color: blue;">+</span>	Ogled vloge	+ <span style="color: blue;">+</span>	Elektronsko vročanje	+ <span style="color: blue;">+</span>	Odločanje	+ <span style="color: blue;">+</span>	Zaključena vloga	+ <span style="color: blue;">+</span>	
The name of the step in the process																					
Elektronsko plačilo																					
Podpisovanje vloge																					
Prilaganje plačila																					
Izbor prejemnika																					
The name of the step in the process	Feature																				
Obrazec	+ <span style="color: blue;">+</span>																				
Ogled vloge	+ <span style="color: blue;">+</span>																				
Elektronsko vročanje	+ <span style="color: blue;">+</span>																				
Odločanje	+ <span style="color: blue;">+</span>																				
Zaključena vloga	+ <span style="color: blue;">+</span>																				
	<table border="1"> <tr><td>Ogled vloge [VNOS] → Obrazec</td><td>X</td></tr> <tr><td>Elektronsko vročanje [VNOS] → Ogled vloge</td><td>X</td></tr> </table>			Ogled vloge [VNOS] → Obrazec	X	Elektronsko vročanje [VNOS] → Ogled vloge	X														
Ogled vloge [VNOS] → Obrazec	X																				
Elektronsko vročanje [VNOS] → Ogled vloge	X																				

Picture 2.6 - Process definition customization

### 2.1.2. Procedure actions customization

Back to the procedure creation administration started in 2.1. Assuming that “*My new process definition*” process definition was selected, the following process definition actions are displayed in the dialog. Button  is used to customize the corresponding process action.

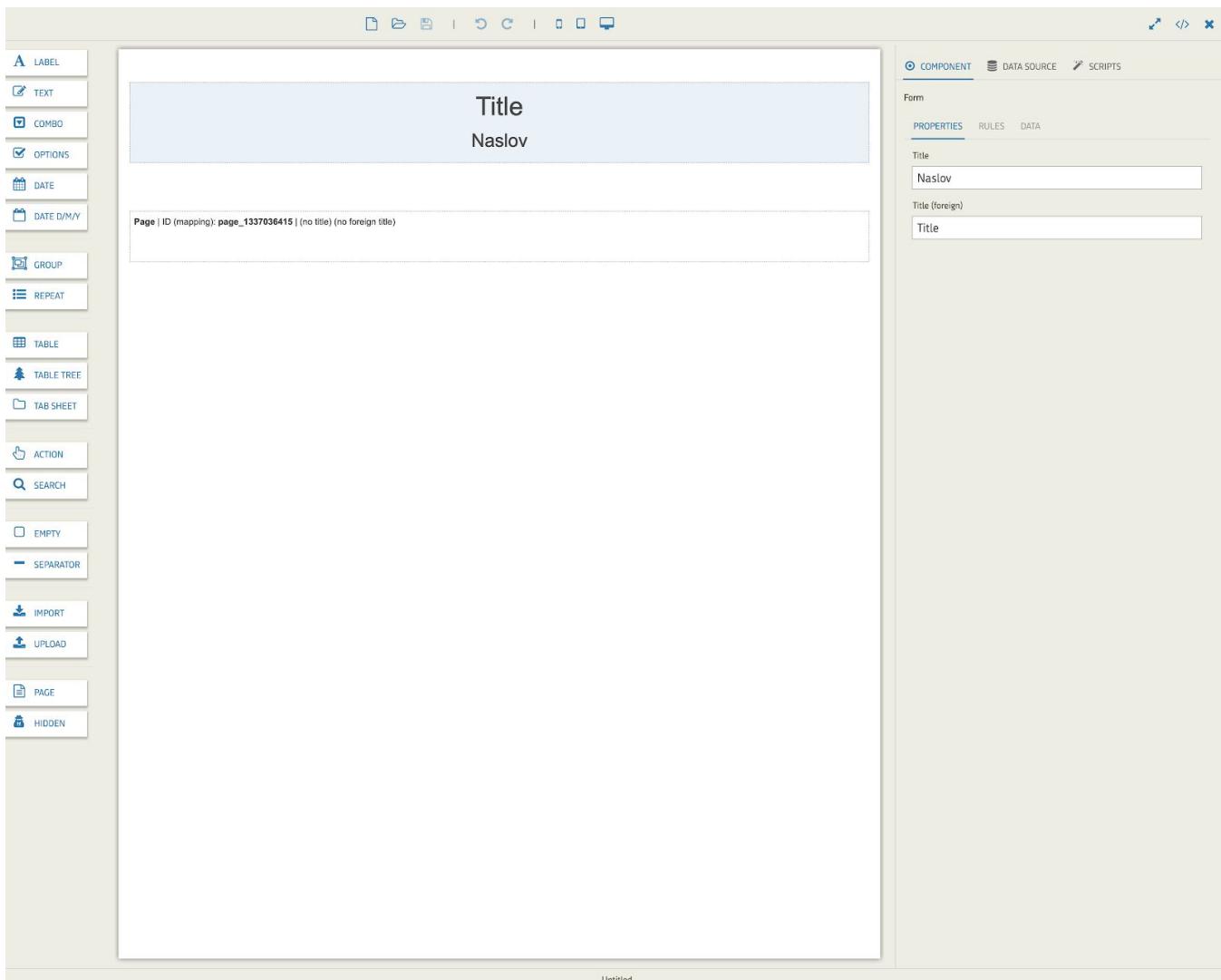
New record

Name: *	Moj novi testni postopek						
Foreign language: *	Angleščina (en)						
Name (foreign): *	My new test procedure						
Competent organization: *	2482762000 - MJU	Prepared by the * organization:	2482762000 - MJU				
Class. category:	0 - DRŽAVNA IN DRUŽBENA UREDITEV						
Class. subcategory:	00 - USTAVNA IN DRŽAVNA UREDITEV						
Classification number: *	000 - Splošno o ustavi in državni ureditvi						
Description:	<p>Roles:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>			Name	Description		
Name	Description						
More info (URL with http://):	www.medijs.si						
More info (foreign) (URL with http://):	www.medijs.si						
Date from:	<input type="text"/>	Immediately active:	<input checked="" type="checkbox"/>				
Process definition: *	My new process definition (2)						
Type of procedure:	Classic						
Process actions:	The name of the step in the process	The order	Feature				
	Obrazec	1					
	Ogled vloge	2					
	Elektronsko vročanje	3					
	Odločanje	4					
	Zaključena vloga	5					

Picture 2.7 - Customization of process definition actions

The most important action customization is the customization of the application form (“*Obrazec*”). It is also the only required customization. Customization of the “*Obrazec*” is de facto creation of a new application form for the particular procedure.

When you click the “*Obrazec*” customization feature button, the following dialog is opened. It is an application form editor with an empty form (empty first page and the title), as shown in picture 2.8.



Picture 2.8 - New application form

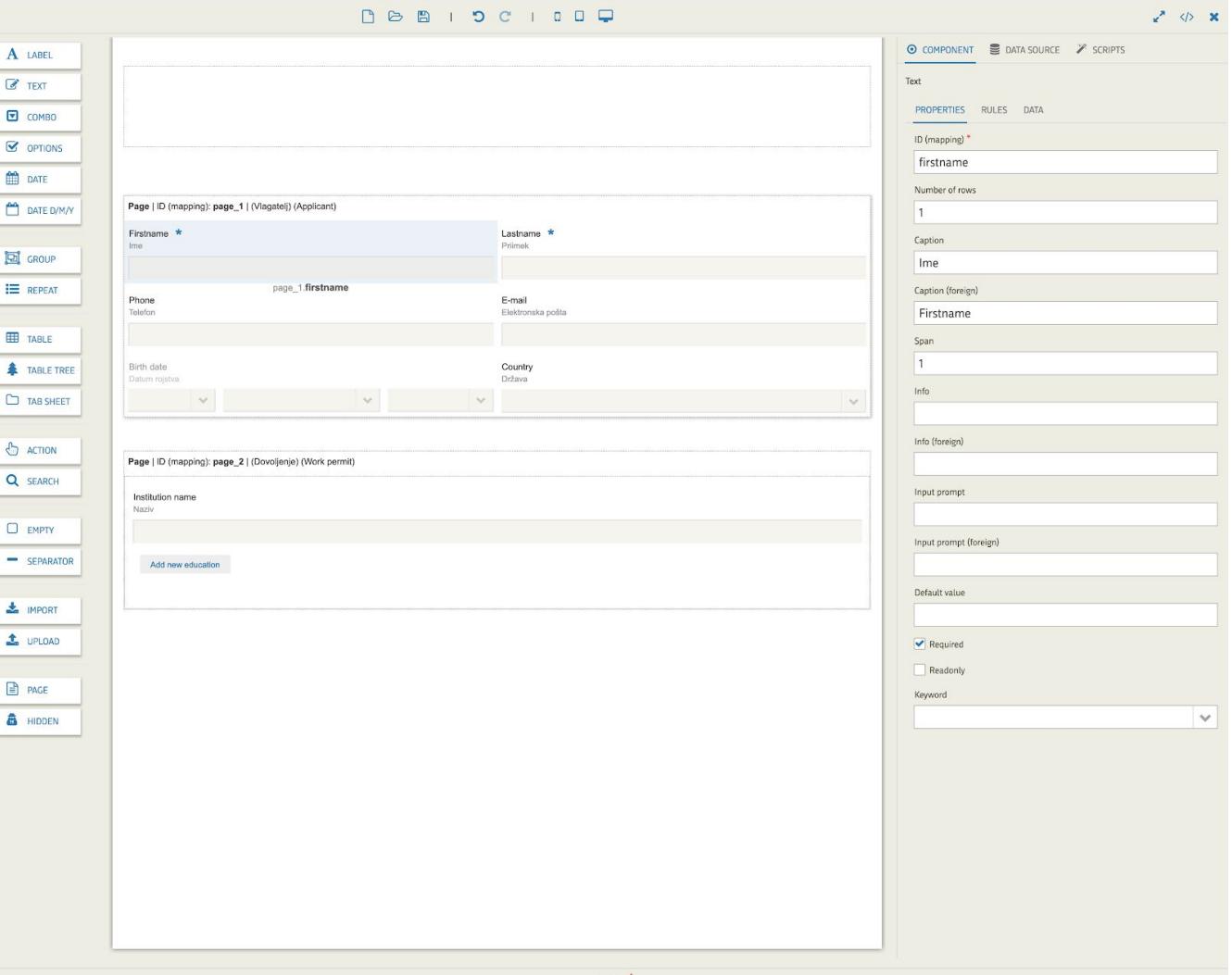
The application form editor consists of five functional sections:

- Left - buttons representing graphical components that can be used to design the application form.
- Middle - a canvas with the graphical components that can be dragged and dropped by the form administrator. It is a WYSIWYG representation of the application form.
- Right - property editor for each GUI component. It is used to customize component's visualization properties, define data gathering and rules, write logic with Groovy scripts.
- Top central - buttons that enable application form preview, simulation and testing. Previews are for mobile, tablet and desktop devices.

- Top right - buttons that enable switching between the canvas mode and the XML mode of the application form representation.

For the purpose of this documentation a very simple application form is created as shown in picture 2.9. It consists of two pages:

- Page 1 named “Applicant” with several simple text fields for entering data (firstname, lastname, phone, email, date\_of\_birth) and a simple combobox for choosing a country from the list of countries. For each component a caption is displayed in both English and Slovenian language.
- Page 2 named “Work permit” with a repeatable component that enables multiple data entry of one or more institution names.



The screenshot shows the MEDIUS application form editor interface. On the left is a toolbar with various icons for creating different components like labels, text fields, dropdowns, etc. The main area is divided into two sections: 'Page | ID (mapping): page\_1 | (Vlagatelj) (Applicant)' and 'Page | ID (mapping): page\_2 | (Dovojenje) (Work permit)'. The 'Applicant' section contains fields for Firstname (Ime), Lastname (Priimek), Phone (Telefon), Email (Elektronska pošta), Birth date (Datum rojstva), and Country (Država). The 'Work permit' section contains a single field for Institution name (Naziv) and a button labeled 'Add new education'. To the right of the form, there is a properties panel titled 'COMPONENT' which includes tabs for 'PROPERTIES', 'RULES', and 'DATA'. Under 'PROPERTIES', the 'ID (mapping)' is set to 'firstname', 'Number of rows' is '1', and 'Caption' is 'Ime'. Other properties like 'Span', 'Info', 'Input prompt', 'Default value', and checkboxes for 'Required' and 'Readonly' are also visible.

Picture 2.9 - Simple application form

In chapter 4, the editor functionality is described in more details for each graphical component separately.

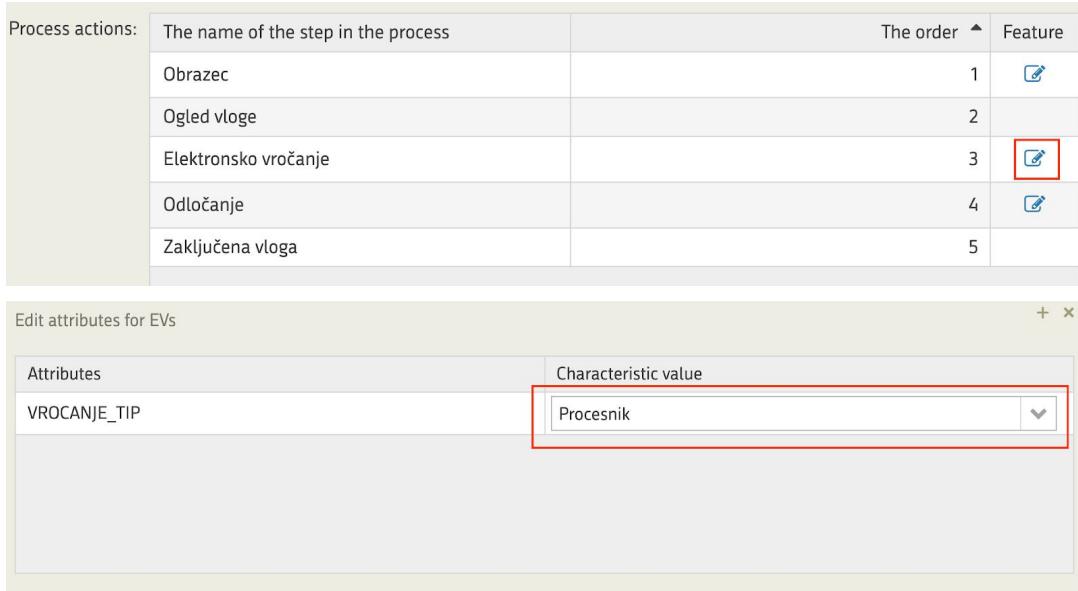
Since each application form is represented by XML, the fastest way to create a new form is to copy/paste one of the existing XML forms into the editor. To do so, switch the presentation mode from the canvas mode to the XML textarea editor mode (button ). Copy/paste the content of the following XML into the XML textarea editor:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<form title="" titleForeign="" id="32160864" xmlns="http://www.mediustsi/mediusforms/schema/Form">
    <pages>
        <page mapping="page_1" title="Vlagatelj" titleForeign="Applicant" id="1209713678">
            <layout columns="2">
                <text rows="1" mapping="firstname" caption="Ime" captionForeign="Firstname" span="1" required="true"
                    readonly="false" id="1159777163"/>
                <text rows="1" mapping="lastname" caption="Priimek" captionForeign="Lastname" span="1" required="true"
                    readonly="false" id="1152181761"/>
                <text rows="1" mapping="phone" caption="Telefon" captionForeign="Phone" span="1" required="false"
                    readonly="false" id="1714006302"/>
                <text rows="1" mapping="mail" caption="Elektronska pošta" captionForeign="E-mail" span="1" required="false"
                    readonly="false" id="1222799962"/>
                <dateDMY mapping="date" caption="Datum rojstva" captionForeign="Birth date" span="1" order="DMY"
                    required="false" readonly="false" monthAsNumber="false" id="470983086"/>
                <combo mapping="country" caption="Država" captionForeign="Country" span="1" required="false" readonly="false"
                    id="1037708688">
                    <items value="SI"/>
                    <items value="I"/>
                    <items value="Hr"/>
                </combo>
            </layout>
        </page>
        <page mapping="page_2" title="Dovoljenje" titleForeign="Work permit" id="1321411541">
            <layout columns="2">
                <repeatable mapping="education" span="2" buttonCaptionAdd="Dodaj nov zapis" buttonCaptionAddForeign="Add new
                    education" buttonCaptionRemove="Odstrani zapis" buttonCaptionRemoveForeign="Remove" hideAddButton="false"
                    hideRemoveButton="false" showAddIcon="false" showRemoveIcon="false" id="586993103">
                    <layout columns="2">
                        <text rows="1" mapping="name" caption="Naziv" captionForeign="Institution name" span="2" required="false"
                            readonly="false" id="1182542352"/>
                    </layout>
                </repeatable>
            </layout>
        </page>
    </pages>
    <scripts/>
</form>
```

Picture 2.10 - XML presentation of the application form

Switch back to canvas mode to see the graphical presentation of the form. Test and try the rendering of the application form by pressing one of the preview buttons . Buttons  and  are used to save and exit the application form. Button  returns back to the procedure metadata dialog shown in picture 2.7.

The type of submission should be set to value “Procesnik”.



Process actions:	The name of the step in the process	The order	Feature
	Obrazec	1	
	Ogled vloge	2	
	Elektronsko vročanje	3	
	Odločanje	4	
	Zaključena vloga	5	

Picture 2.11 - Submission type should be set to “Procesnik”

Button   saves the procedure and exits the procedure’s dialog. The newly created procedure appears in the table view as shown in picture 2.12.



PROCEDURES						SEARCH	MONITOR	STATISTICS
Name:	 					    		
Name (foreign):	Procedure	Tag	Name	Resp. org.	Date from	Date to	Status	ID procedure
	6767	2482762000/03	Moj novi testni postopek	MJI	13.03.2019	01.01.3000	In Edit	6768

Picture 2.12 - Quick searching by using the procedure name

Button  makes a procedure visible on the Jedro-front-frame portal.

### 3. Jedro-front and Jedro-front-frame

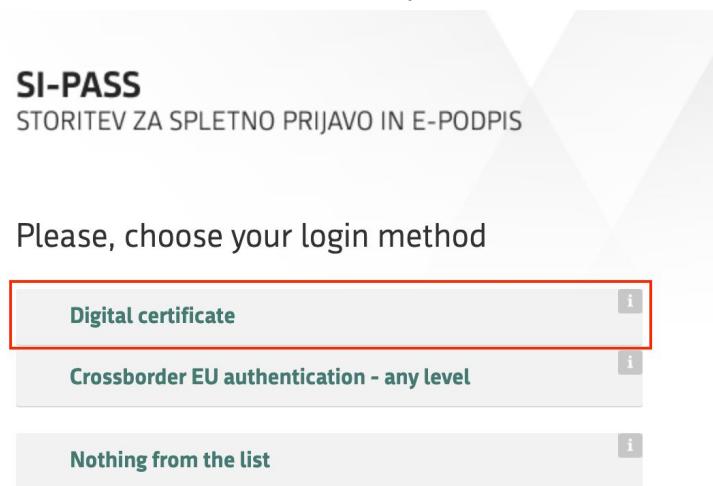
Published procedures can be visualized and used on any 3rd party portals or HTML pages using the CORS techniques of sharing information rendered by WEB systems of different origin. Jedro-front is one of the IS EPS microservices that enables cross origin sharing of procedure visualization. Jedro-front runs the Process State Engine that controls the running of the procedure's states and renders these states as HTML pages. Since Jedro-front provides the API and the JavaScript library for CORS, Jedro-front can be embedded into 3rd party HTML pages so that the running of the Process State Engine is visualized inside that 3rd party HTML pages.

Jedro-front-frame is a simple portal that uses the provided Jedro-front's CORS JavaScript API and library to embed the running of Jedro-front's Process State Engine into its HTML pages.

Jedro-front-frame is available at:

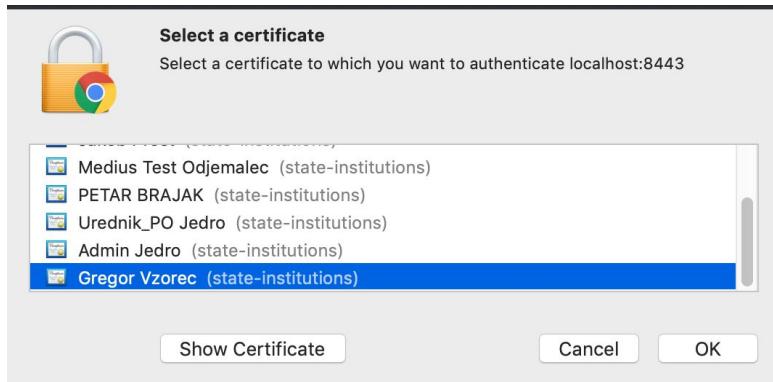
- Medius (development environment): <https://mju-jep.test.mediustest.si/?lang=en>. Use the supplied digital certificate named “*Gregor Vzorec*”.
- Ministry of Public Administration (testing): <https://jep-test.gov.si/?lang=en>. Use the supplied digital certificate named “*Jure Prošč*”.

The URL is redirected to the Central Authentication System.



Picture 3.1 - SI-PASS authentication

For accessing <https://mju-jep.test.mediustest.si/?lang=en> use the “*Gregor Vzorec*” certificate, as shown in picture 3.2.



Picture 3.2 - From authentication

The landing page of Jedro-front-frame is a dashboard that displays all previously created procedures by the logged-in user.

## EPS - Electronic Procedure Services

### Dashboard

Welcome,

#### Company is not selected

You are currently submitting applications as a natural person. If you want to submit applications on behalf of a legal entity, please select the desired entity from the menu at the top of the page.

Find and file a request

🔍

#### Requests in progress

##### Test - vključitev slike

Application identifier: 2482762000/03-101-1-3/2019  
Last saved on: 26.02.2019

[Open request](#) [Delete request](#) [Show details](#)

##### Test - vključitev slike

Application identifier: 2482762000/03-101-1-1/2019  
Last saved on: 26.02.2019

[Open request](#) [Delete request](#) [Show details](#)

##### Craft permit - without sign and pay

Application identifier: 2482/03-313-13-9615/2019  
Last saved on: 15.02.2019

[Open request](#) [Delete request](#) [Show details](#)

##### TOOP - The Once-Only Principle

Application identifier: 1685325000/03-000-3-57/2018  
Last saved on: 07.11.2018

[Open request](#) [Delete request](#) [Show details](#)

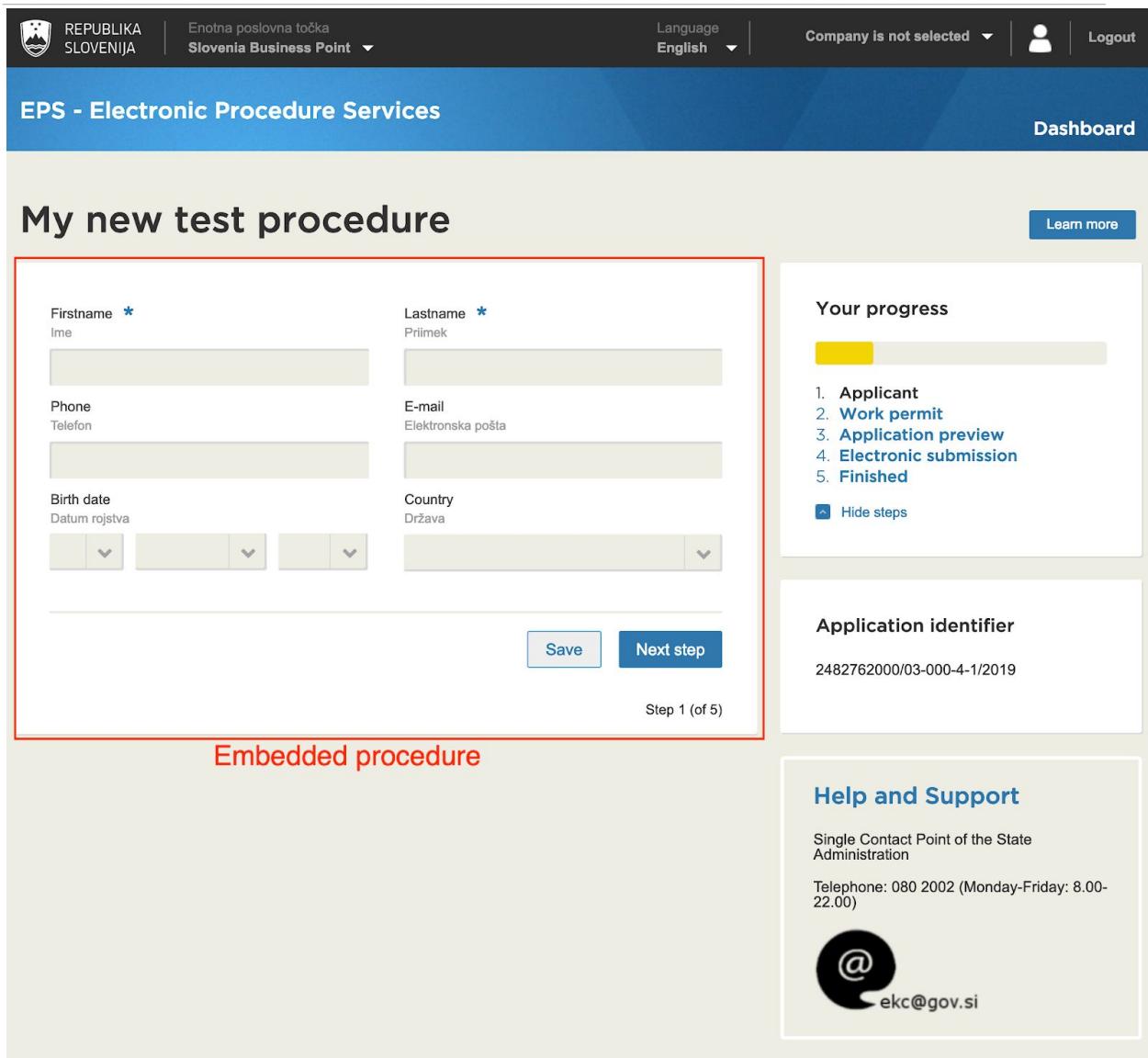
#### Help and Support

Single Contact Point of the State Administration  
Telephone: 080 2002 (Monday-Friday: 8.00-22.00)



Picture 3.3 - Jedro-front-frame

To create a new procedure use the “*Find and file a request*” search component. Type any characters to find a procedure name you wish to start using it. For example “*My new test procedure*” is a testing procedure created in chapter 2. By selecting the procedure name, a new procedure is instantiated in the Process State Engine of the Jedro-front. The procedure rendering is embedded into the HTML iframe tag of Jedro-front-frame as shown in picture 3.4.



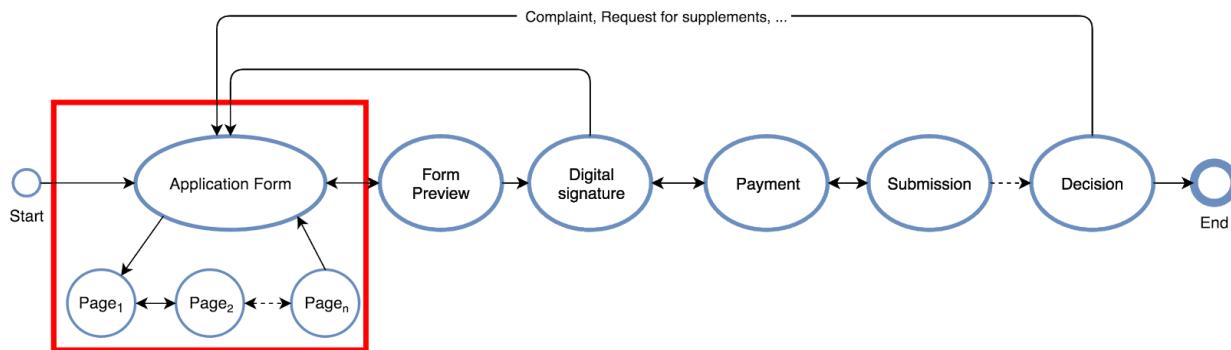
The screenshot shows the EPS - Electronic Procedure Services dashboard. At the top, there are navigation links for REPUBLIKA SLOVENIJA, Enotna poslovna točka Slovenia Business Point, Language English, Company is not selected, and Logout. Below the header, the title "EPS - Electronic Procedure Services" and "Dashboard" are displayed. A large central area is titled "My new test procedure". It contains a form with fields for Firstname (Ime), Lastname (Priimek), Phone (Telefon), E-mail (Elektronska pošta), Birth date (Datum rojstva), and Country (Država). Buttons for "Save" and "Next step" are visible. A progress bar on the right indicates Step 1 (of 5) is completed. To the right of the form, a "Your progress" section lists steps: 1. Applicant, 2. Work permit, 3. Application preview, 4. Electronic submission, 5. Finished, with an option to "Hide steps". Below the form, the text "Embedded procedure" is highlighted in red. To the right, sections for "Application identifier" (2482762000/03-000-4-1/2019) and "Help and Support" (Single Contact Point of the State Administration, Telephone: 080 2002 (Monday-Friday: 8.00-22.00), email icon with ekc@gov.si) are shown.

Picture 3.4 - Embedded procedure into the dashboard

On the right side of the dashboard there is a list of all procedure steps (*Your progress*). This list is provided by the Jedro-front via CORS Javascript API. Furthermore, Jedro-front provides an API to fetch the procedure name which is displayed at the top of dashboard. The information about page height is also provided by Jedro-front to Jedro-front-frame via CORS based JavaScript API.

## 4. Application Form Editor

This chapter describes the functionality of the application form editor. Application form editor is the most important tool of the system. As shown in picture 4.1, the application form editor is used to customize the action node “Obrazec” (Application Form).



The application form is a graphical WYSIWYG HTML based editor that enables administrators to design application forms using the following graphical component.

- Layout components
  - **PAGE** - The primary container component, onto which all other components must be placed. The applicant can only navigate to the next page when he has filled out all the required fields on the current page.
  - **GROUP** - This component acts as a container into which other components can be nested. It can be configured to have any number of columns, allowing for better control over the layout of components on the form. Used for creating logical groups of components, that can later be positioned, hidden, disabled, etc.
  - **REPEAT** - This component is a group of repeatable components. Used for providing the applicant with a way to repeat information of same kind. For example, to add multiple billing addresses.
  - **TABSHEET** - This is a container of components inside a tabsheet. This component allows switching between configurable tabs, each containing its own layout, onto which other components can be placed. Used for segregation of components into groups, which the applicant can switch between and fill out in any order using the tabs at the top of this component.
- Input components
  - **LABEL** - This component displays text, which can be formatted using HTML. Used to display (emphesys) relevant information to the applicant.

- **TEXT** - This component allows input of text values. The size of the field is configurable by changing the number of lines in the property. Used to get text input from the applicant, such as names, surnames, addresses. It is also used for mapping data from external data source.
- **COMBO** - This component allows the applicant to choose an item from a predefined set of options, presented in the form of a drop down menu that also supports quick filtering. Used for representing choices, for example, nationality, education, ...
- **OPTION** - This component allows the user to choose one of a predefined set of options. The options can either be presented as radio buttons (one option to be selected), or as checkboxes (multi selection). A common and powerful use for the Options component is triggering rules and/or scripts based on the users choice. For example, hiding or showing a group component, if the user selects "yes" or "no", etc.
- **DATE** - This component allows for quick and easy date selection, using a popup visual calendar, that opens when the calendar icon on the field is clicked.
- **DATE D/M/Y** - This component allows for date selection with separate fields for the day, month and year. The field order and separator are both configurable. Used for choosing dates in a specific order, or with a specific separator.
- **TABLE** - This component displays data in a table. The number of columns and their sizes are all configurable. By default, the component is read only, but can also be configured to enable single or multiple row selection. Used for displaying data in a table, and optionally, enabling single or multiple row selection.
- **TABLE TREE** - This component displays data in a table with expandable/collapsible rows replicating the tree structure of its items. The number of columns and their sizes are all configurable. By default, the component is read only, but can also be configured to enable single or multiple row selection. Used for displaying data in a table, where each row can have multiple sub rows, and optionally, enabling single or multiple row selection.
- **ACTION** - A button which can be configured to trigger rules, scripts and data sources.
- **SEARCH** - This component is a composite component of the Text and Action components.
- **EMPTY** - This is an empty component and is used to add white/blank space to the form. Used for adding white space to the form.
- **SEPARATOR** - This component consists of a label underscored by a horizontal line. The labels value can be left blank, if only the horizontal line is needed. Used for improving visual clarity by adding visual separation between components.
- **UPLOAD** - This component enables uploading of files/attachments to the form by the applicant. Only files of the configured allowed mimetypes will be allowed.

Used for acquiring required applicant's documents, for example, identification or other legal documents, education diploma...

- Special components

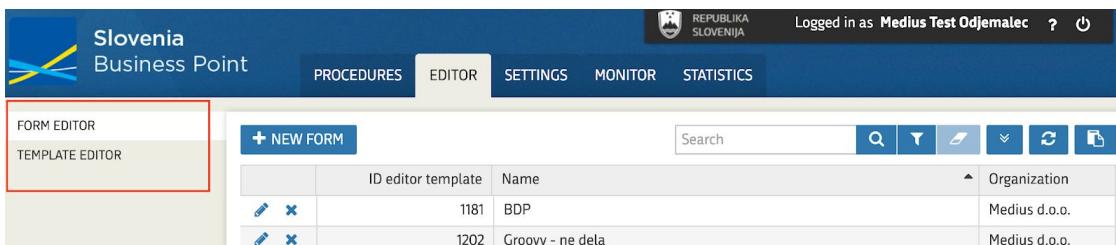
- **IMPORT** - This component is used to import previously saved custom components. After clicking on the import button and selecting a saved custom component, this component will be replaced with the selected custom component.
- **HIDDEN** - This component consists of a field, which is invisible to the applicant when filling out the form, but available for any background data processing if needed.

In general, the editor for building the application forms is used by the administrator during the phase of procedure creation as demonstrated in chapter 2.1. When the feature button of the process action "Obrazec" (Application Form) is pressed, the application form editor opens.

Process actions:	The name of the step in the process	The order	Feature
	Obrazec	1	
	Ogled vloge	2	
	Elektronsko vročanje	3	
	Odločanje	4	
	Zaključena vloga	5	

The newly created application form is associated with the procedure and cannot be reused by other procedures. Since the application form is just an XML file, the administrators can maintain their own list of application form XML files and copy/paste one of the XML files into the editor. But in general, an application form is designed during the phase of procedure creation.

However, it is possible to create an application forms independently from the procedure creation. This is useful when creating the application forms in advance, testing and trying different form features, proof-of-concepts, etc. To create such a form independently from the procedure, go to menu EDITOR → FORM EDITOR.



The screenshot shows the MEDIUS Business Point application interface. At the top, there is a navigation bar with the Slovenia Business Point logo, the REPUBLIKA SLOVENIJA coat of arms, and a user logged in as 'Medius Test Odjemalec'. Below the navigation bar, there are tabs for PROCEDURES, EDITOR (which is currently selected), SETTINGS, MONITOR, and STATISTICS. On the left side, there is a sidebar with buttons for 'FORM EDITOR' (highlighted with a red box) and 'TEMPLATE EDITOR'. The main content area has a header with a '+ NEW FORM' button and a search bar. Below the header, there is a table with columns for 'ID editor template', 'Name', and 'Organization'. Two rows are visible in the table:

ID editor template	Name	Organization
1181	BDP	Medius d.o.o.
1202	Groovy - ne dela	Medius d.o.o.

The forms created that way can be reused in the phase of procedure creation by selecting one of them. The button “Load” shown in the following picture, opens the list of forms that had been created in the menu EDITOR → FORM EDITOR. A selected form is loaded into the editor.

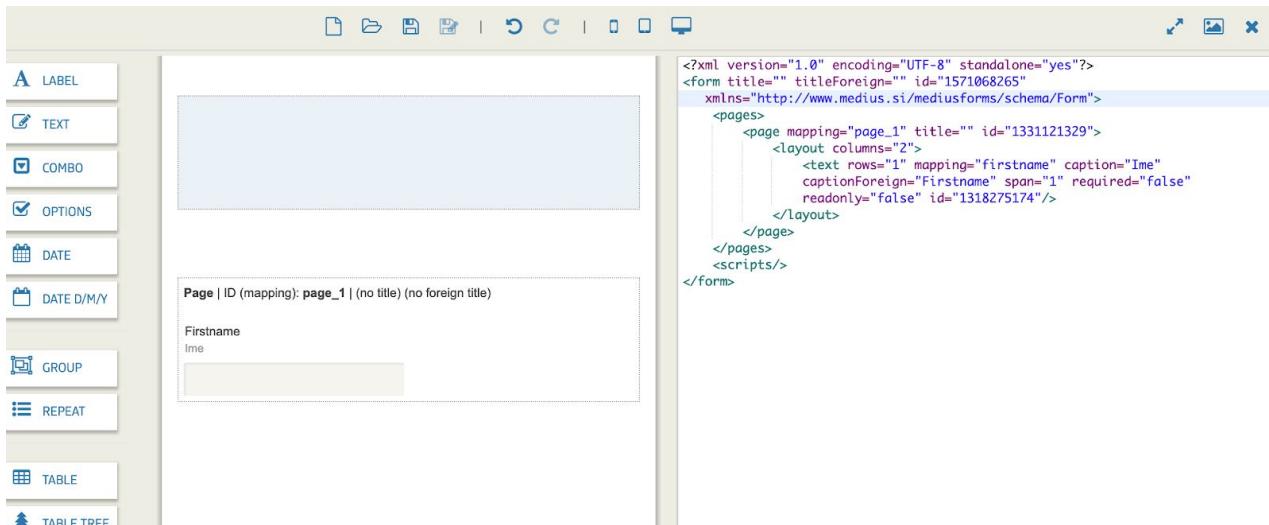


Furthermore, it is also possible to create the parts of the application forms which are represented as group components (Group, Repeatable, Tabsheet). Use menu EDITOR → TEMPLATE EDITOR for this functionality. Button IMPORT in the application editor enables importing one of the templates created in this menu.

## 4.1. XML presentation

Each application form is represented by the corresponding XML file as shown in picture 2.10.

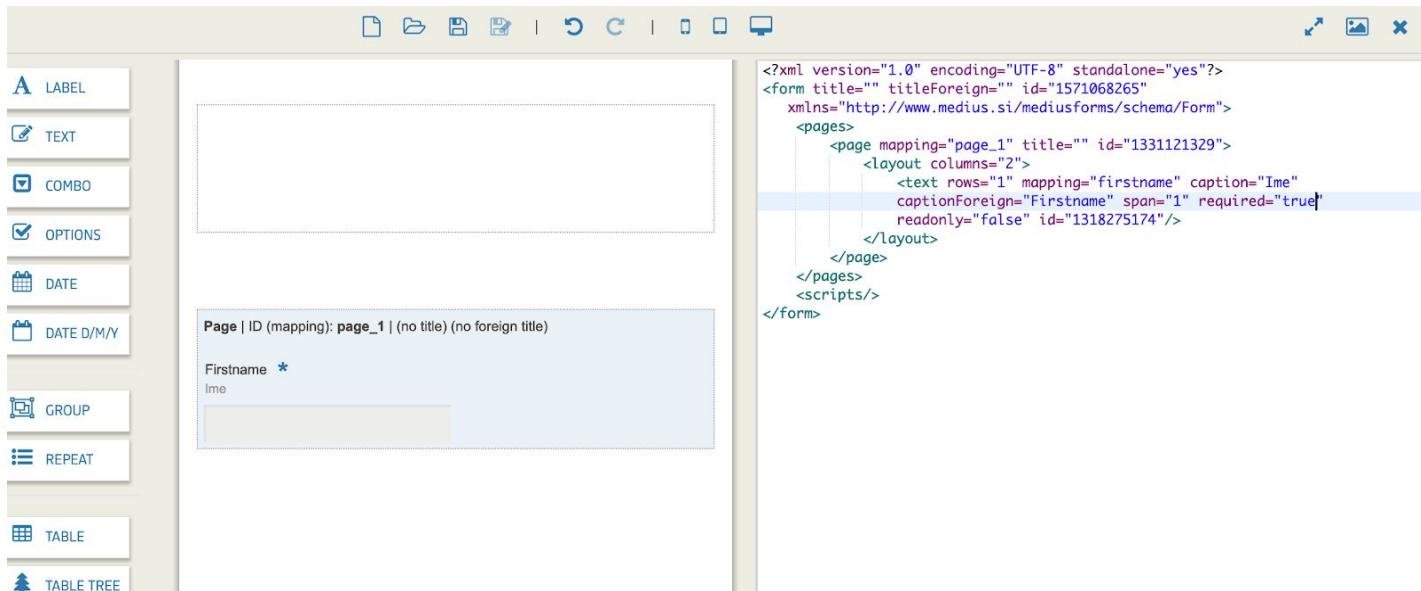
Switch buttons  and  toggle between the property editor and the XML presentation editor. For example, in picture 4.1 and 4.2 there is a page named “page\_1” (*mapping*=“page\_1”) and the input text field named “firstname” (*mapping*=“firstname”) on the canvas.



Picture 4.1 - XML presentation of the application form

If you type “true” instead of “false” for the property *required*=“false” in the XML editor, the canvas presentation is immediately refreshed and the text field displays the required sign (\*). And vice

versa. If the property is changed on the property editor of the canvas, the change is automatically recorded in the XML.

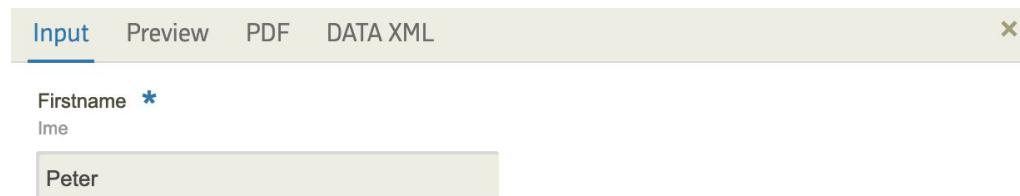


Picture 4.2 - Automatic canvas refresh when XML is changed

## 4.2. Previews

Administrators can simulate and test the behavior of application forms by using the preview buttons . The preview has 4 different application form visualizations.

- Input mode - as seen by the applicant when filling the form. This mode is used for simulating and testing form data entry.



Picture 4.3 - Input mode

- Preview mode - application form displayed as HTML.
- PDF mode - application form displayed as PDF.

- Data XML - display of all filled data. Data XML is a name/value tuple of all application form input fields. Can be used by the competent organizations to process submitted form. The competent organization receives a digitally signed PDF file and data XML file for each application form.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<formdata lastValidPage="page_1" xmlns="http://www.mediustech.com/mediusforms/schema/FormData">
    <item mapping="page_1">
        <data name="firstname" value="Peter" verified="false"/>
    </item>
</formdata>
```

Picture 4.4 - Form meta data represented as name/value tuple

## 4.3. Graphical components

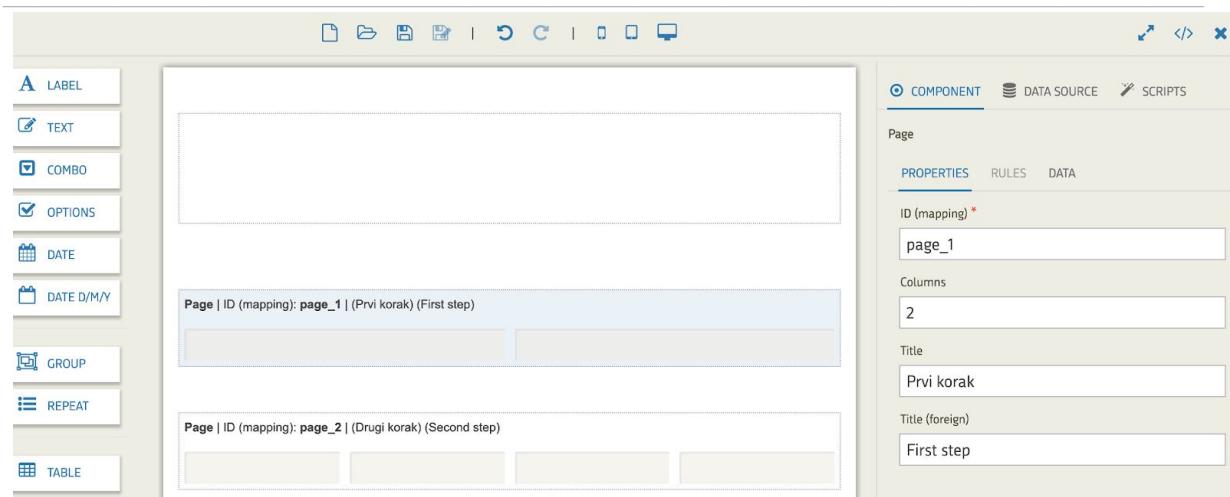
This section describe the features of the components used for building application forms. It is not an exhaustive description, but just the demonstration of the coolest components and the respective features.

### 4.3.1. Layouts

Page, Group, Repeat and TabSheet are four components that can be used as containers for other components. The primary role of these components is to set up the layout for other components by defining the number of columns per page, per group, or per tabsheet. Furthermore, these layout components are used for logical metadata grouping.

#### 4.3.1.1. Page

This component acts as a container into which other components can be nested. It can be configured to have any number of columns, allowing for better control over the layout of components in the form. Used for creating logical groups of components, that can later be positioned, hidden, disabled, etc. together, by referring to their parent group component. Picture 4.5 shows two pages in the form, the first page having layout of two columns, the second page having a layout of four columns.



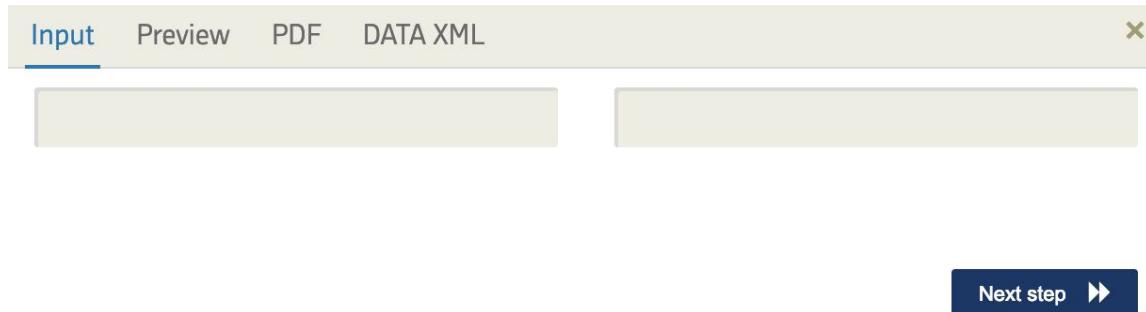
The screenshot shows the MEDIUS application builder interface. On the left, there is a toolbar with icons for various components: Label, Text, Combo, Options, Date, Date D/M/Y, Group, Repeat, and Table. The main area consists of two stacked panels. The top panel is labeled "Page | ID (mapping): page\_1 | (Prvi korak) (First step)" and contains a single large rectangular component. The bottom panel is labeled "Page | ID (mapping): page\_2 | (Drugi korak) (Second step)" and contains four smaller rectangular components arranged horizontally. To the right of these panels is a sidebar titled "Page" with tabs for "COMPONENT", "DATA SOURCE", and "SCRIPTS". Under the "COMPONENT" tab, there are sections for "PROPERTIES", "RULES", and "DATA". In the "PROPERTIES" section, the "ID (mapping)" field is set to "page\_1". The "DATA" section includes fields for "Columns" (set to "2"), "Title" ("Prvi korak"), and "Title (foreign)" ("First step").

Picture 4.5 - Page properties

Use the right mouse click to cut/copy/paste/delete any component on the canvas.

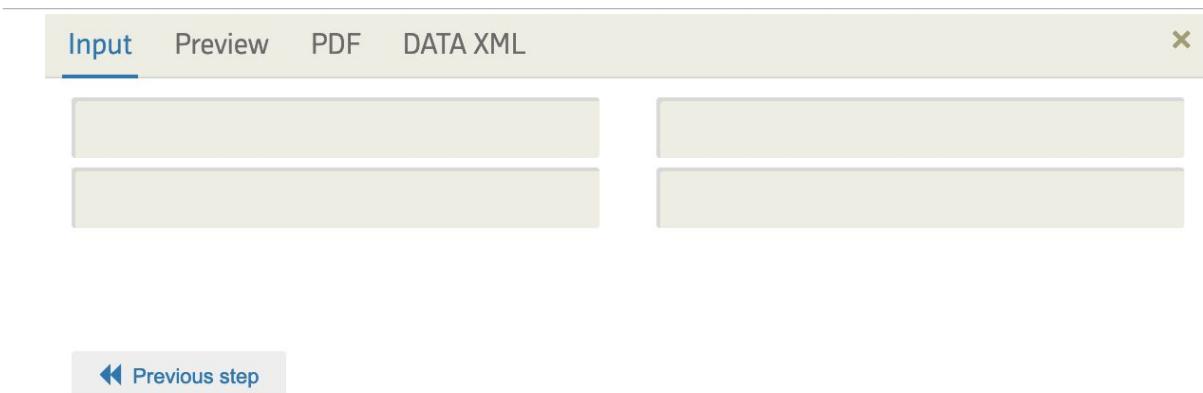


Administrators can test and try the page behaviour by previewing the form. For example, the application form defined in picture 4.5 displays two pages, each one with its own layout.



The screenshot shows the preview mode of the application. At the top, there is a navigation bar with tabs: "Input" (which is active), "Preview", "PDF", and "DATA XML". Below the navigation bar, there are two large, empty rectangular boxes representing the two pages from Picture 4.5. At the bottom right, there is a blue button labeled "Next step" with a right-pointing arrow.

Picture 4.7 - Page 1 with layout=2

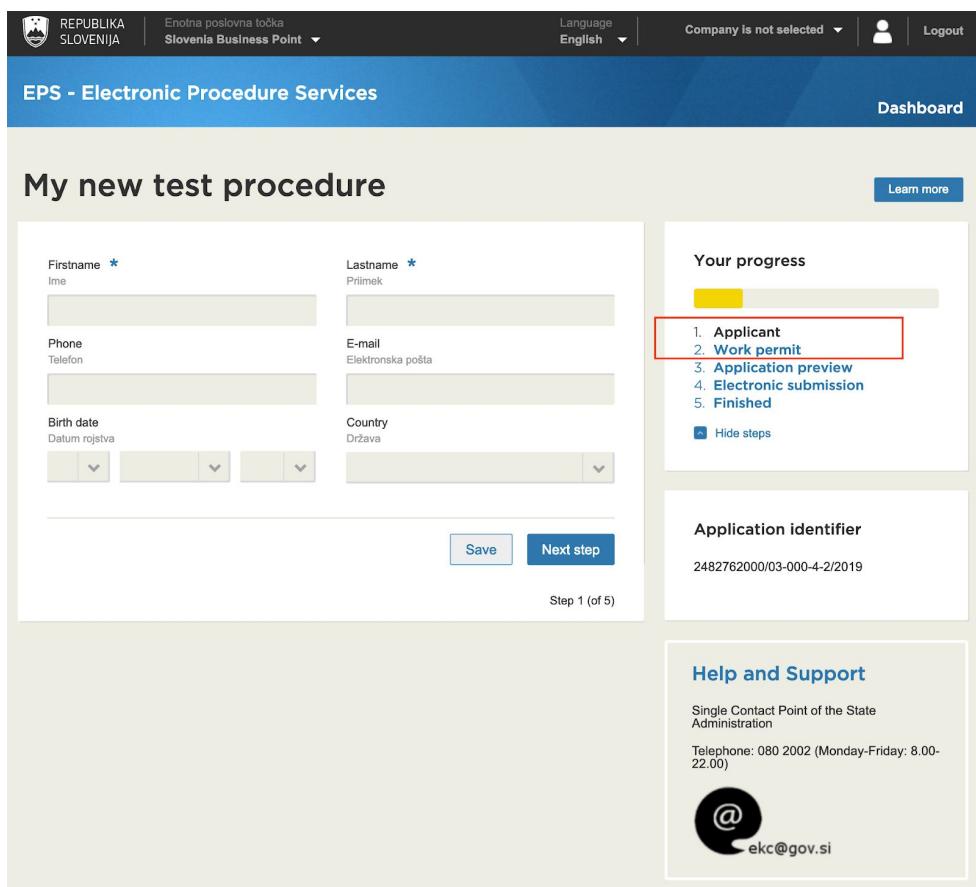


Input   Preview   PDF   DATA XML

◀ Previous step

Picture 4.7 - Page 2 with layout=4

The page title is used by the rendering portal to name the steps of the application form. For example, the first two steps of the procedure “*My new test procedure*” are two pages in the application form (Applicant, Work permit). Other procedure step names are from names of nodes of the process definition.



REPUBLICA SLOVENIJA | Enotna poslovna točka Slovenia Business Point | Language English | Company is not selected | Logout

EPS - Electronic Procedure Services | Dashboard

## My new test procedure

Learn more

Firstname *	Lastname *
Ime	Primek
Phone	E-mail
Telefon	Elektronska pošta
Birth date	Country
Datum rojstva	Država

Save   Next step

Step 1 (of 5)

Your progress

- 1. Applicant
- 2. Work permit
- 3. Application preview
- 4. Electronic submission
- 5. Finished

Hide steps

Application identifier

2482762000/03-000-4-2/2019

Help and Support

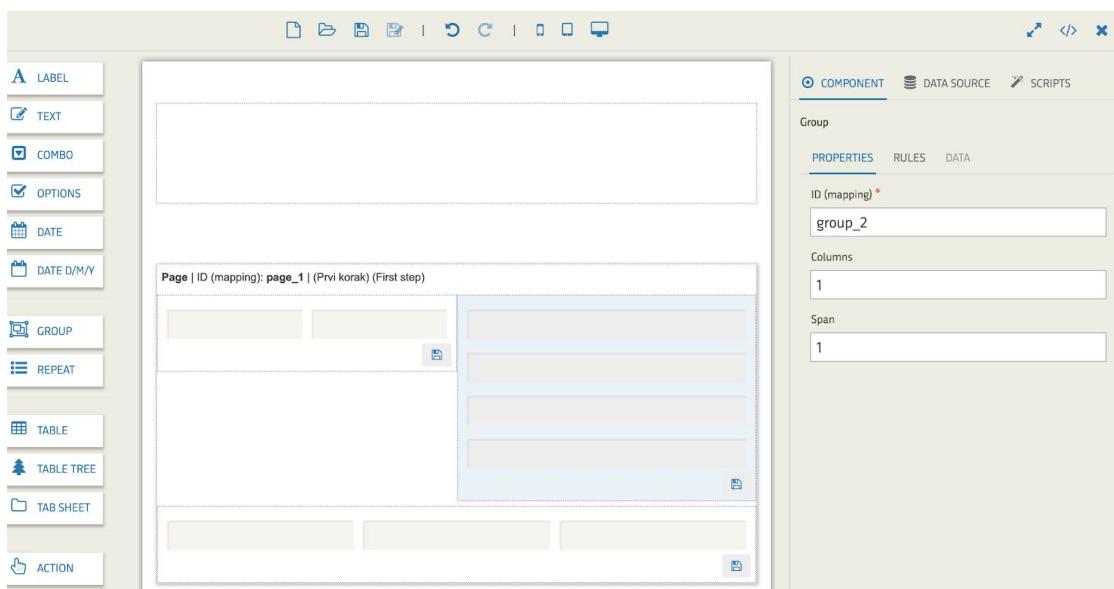
Single Contact Point of the State Administration  
Telephone: 080 2002 (Monday-Friday: 8.00-22.00)

@ [ekc@gov.si](mailto:ekc@gov.si)

Picture 4.6 - Page titles displayed as process steps of the procedure

#### 4.3.1.2. Group

Group is a container of components having similar logical behaviour. This component acts as a container into which other components can be nested. It can be configured to have any number of columns, allowing for better control over the layout of components in the form. Used for creating logical groups of components, that can later be positioned, hidden, disabled, etc. together by referring to their parent group component. For example, the following page of layout=2 contains three groups.



Picture 4.8 - Page with 3 group components

The first group having layout two (columns=2), the second group having layout one (column=1) and the third group having layout three (columns=3) and span=2. Spanning (default is 1) defines how components span regarding the parent's layout. Since a page in picture 4.8 has defined layout of 2, the third group is spread horizontally throughout page layout because of span=2.

Button  is used for saving the component as a template for other application forms. The template can be imported by button  IMPORT .

To simulate/test a group behaviour use one of the preview buttons.



Picture 4.9 - Groups with different layouts

Groups are also used for unambiguous naming of the form components as shown in the following picture representing the data XML. Although three components have the same name "id1" they can be distinguished by the parent group names: "group\_1.id1", "group\_2.id1", "group\_3.id1".

```

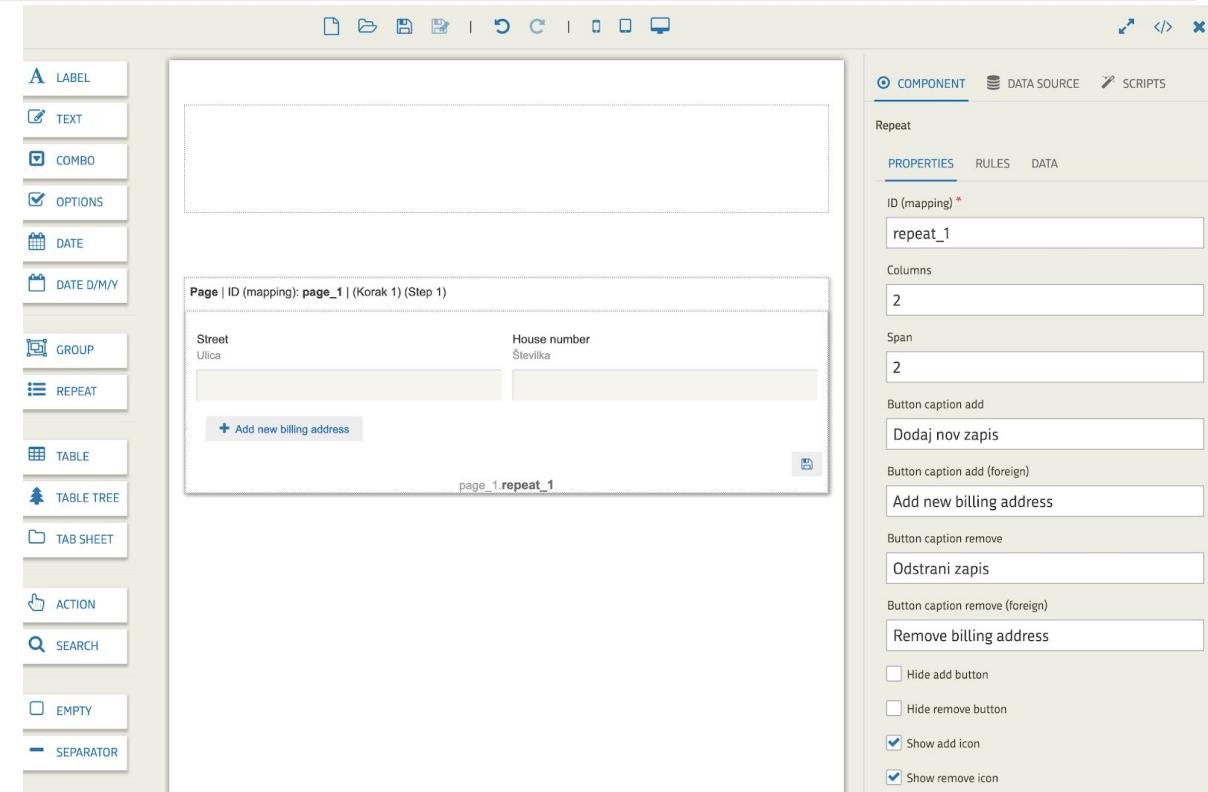
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<formdata lastValidPage="page_1" xmlns="http://www.mediustech.com/mediusforms/schema/FormData">
    <item mapping="page_1">
        <item mapping="group_1">
            <data name="id1" value="" verified="true"/>
            <data name="id2" value="" verified="true"/>
        </item>
        <item mapping="group_2">
            <data name="id1" value="" verified="true"/>
            <data name="id2" value="" verified="true"/>
            <data name="id3" value="" verified="true"/>
            <data name="id4" value="" verified="true"/>
        </item>
        <item mapping="group_3">
            <data name="id1" value="" verified="true"/>
            <data name="id2" value="" verified="true"/>
            <data name="id3" value="" verified="true"/>
        </item>
    </item>
</formdata>

```

Picture 4.10 - Unambiguous name/value tuples

#### 4.3.1.3. Repeat

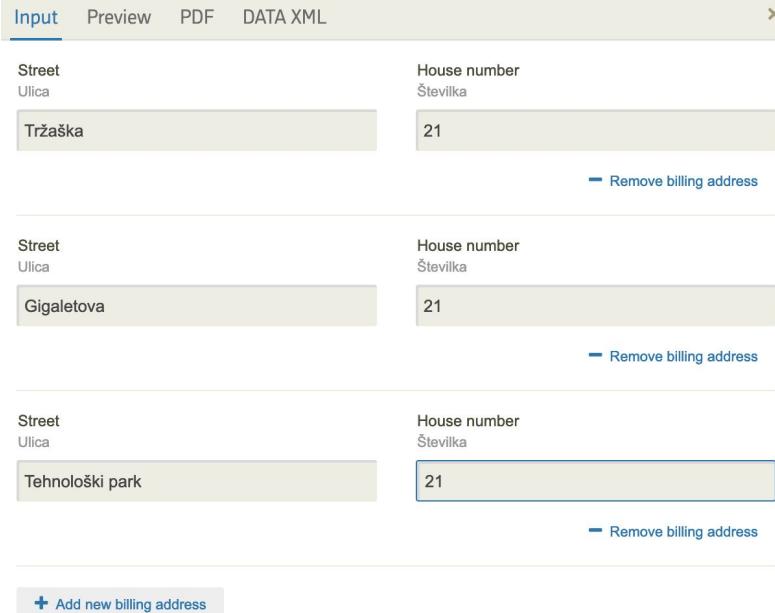
A Repeat component is a special kind of a group component. It acts as a container into which other components can be nested. However, this container can be duplicated or removed by the applicant according to his/her needs by using the add and remove buttons. The component is used for providing the applicant with a way to repeat a group of components. For example, to add multiple billing addresses.



The screenshot shows the MEDIUS application's configuration interface. On the left, a toolbar provides various components: Label, Text, Combo, Options, Date, Date D/M/Y, Group, Repeat, Table, Table Tree, Tab Sheet, Action, Search, Empty, and Separator. The main area displays a configuration for a 'Repeat' component. The preview window shows a form with a header 'Page | ID (mapping): page\_1 | (Korak 1) (Step 1)'. It contains two address fields: 'Street' (Ulica) and 'House number' (Številka). Below these is a button '+ Add new billing address'. The configuration panel on the right shows properties for the repeat component, including the ID 'repeat\_1', a column count of '2', and a span of '2'. It also includes sections for 'Button caption add' ('Dodaj nov zapis'), 'Button caption add (foreign)' ('Add new billing address'), 'Button caption remove' ('Odstrani zapis'), 'Button caption remove (foreign)' ('Remove billing address'), and checkboxes for hiding add and remove buttons and showing icons.

Picture 4.11 - Repeat component

The resulting form enables an application to input multiple information of the same type:



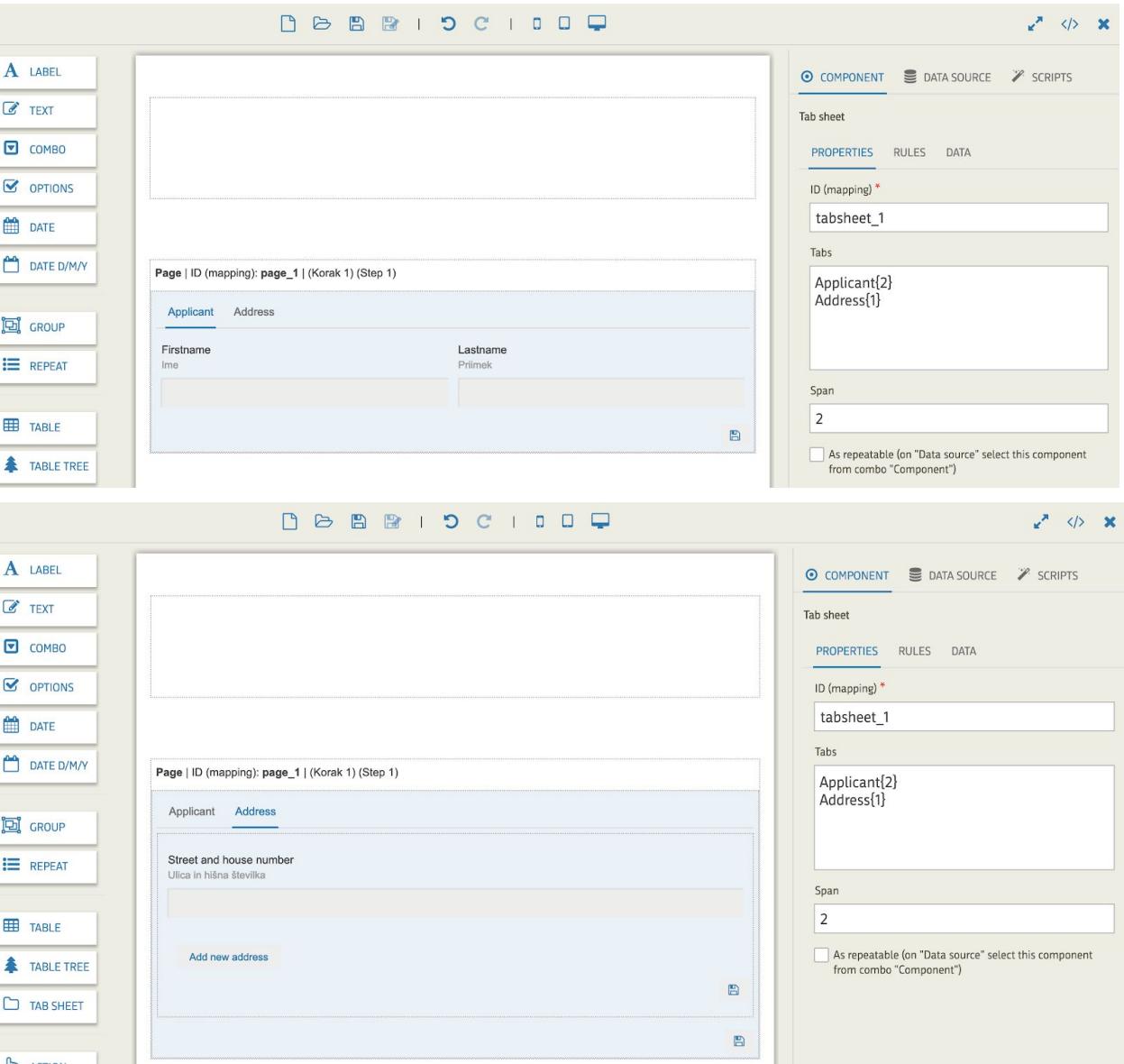
The screenshot shows the final form generated by the repeat component. It consists of three identical address entry rows. Each row has 'Street' (Ulica) and 'House number' (Številka) fields. The first row contains 'Tržaška' and '21'. The second row contains 'Gigaletova' and '21'. The third row contains 'Tehnološki park' and '21'. Below each row is a 'Remove billing address' link. At the bottom left is a '+ Add new billing address' button.

Street	House number
Tržaška	21
Gigaletova	21
Tehnološki park	21

Picture 4.12 - Repeat component in action

#### 4.3.1.4. Tabsheet

Tabsheet is a container of components visualised as a tabsheet. This component allows switching between configurable tabs, each containing its own layout, onto which other components can be placed. Used for segregation of components into groups, which the applicant can switch between and fill out in any order using the tabs at the top of this component. In picture 4.14 a Tabsheet component is shown having 2 tabs, first tab named Applicant with layout=1 and second tab named Address with layout=1. Each tab can contain different components, layouts, spannings, etc. For example, the first tab displays two input fields, the second tab Address displays a repeatable component with one input field.



The image consists of two vertically stacked screenshots of a form builder application interface. Both screenshots show a central workspace for designing a form, a left sidebar with component icons, and a right panel for configuring the selected component.

**Screenshot 1 (Top):** The central workspace shows a Tabsheet component with two tabs: "Applicant" and "Address". The "Applicant" tab contains two input fields: "Firstname" and "Lastname". The "Address" tab is currently selected and contains a "REPEAT" component with a single input field. The right panel shows the component configuration for "tabsheet\_1". Under the "Tabs" section, it lists "Applicant[2]" and "Address[1]". A "Span" field is set to "2". A note at the bottom says: "As repeatable (on "Data source" select this component from combo "Component")".

**Screenshot 2 (Bottom):** Similar to the first screenshot, but the "Address" tab now contains a "TABLE" component instead of a "REPEAT" component. The right panel shows the component configuration for "tabsheet\_1". Under the "Tabs" section, it lists "Applicant[2]" and "Address[1]". A "Span" field is set to "2". The same note about repeatable components is present.

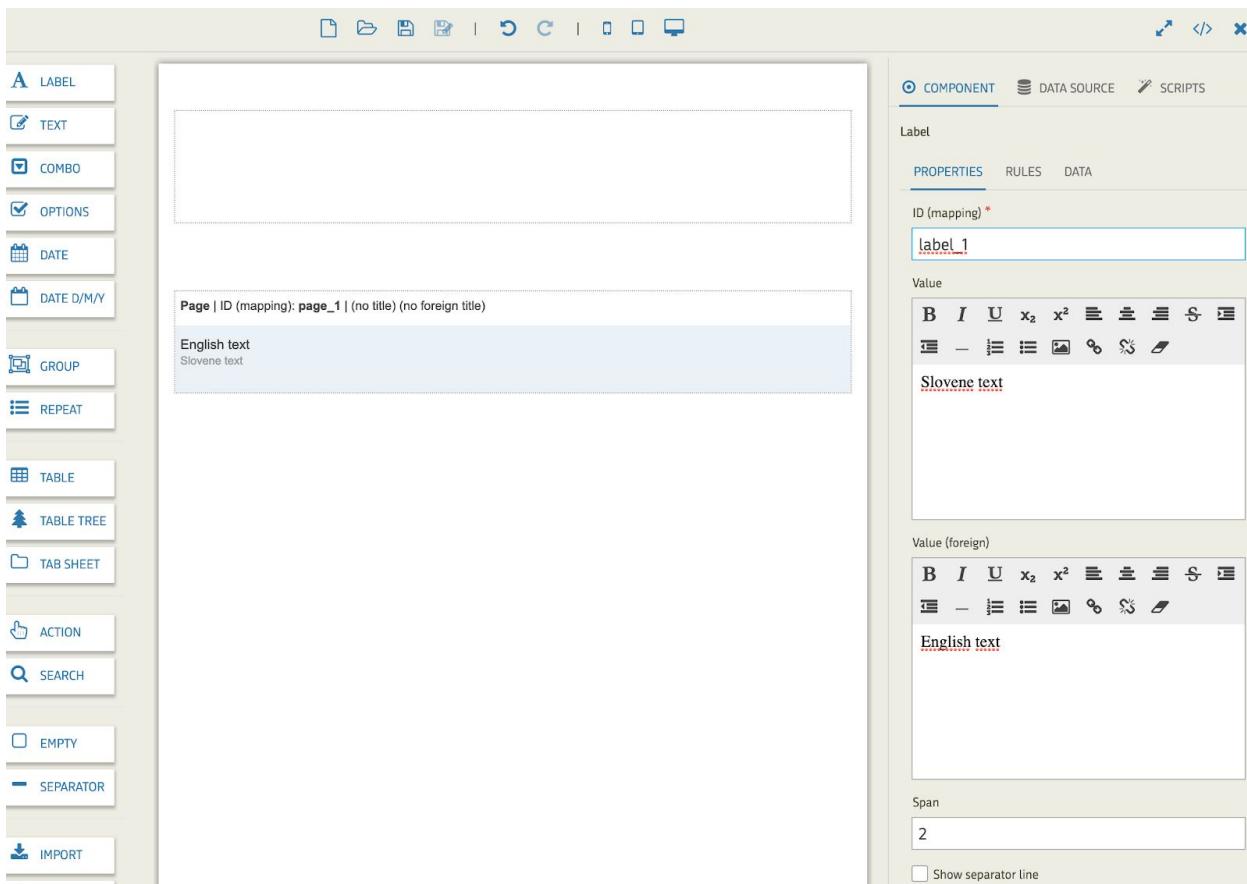
Picture 4.14 - Tabsheet with two different tabs

### 4.3.2. Input components

This chapter demonstrates the behavior of some of the most important graphical input components.

#### 4.3.2.1. Label

Label component displays text, which can be formatted using HTML. It is used to display (emphesys) relevant information to the applicant.



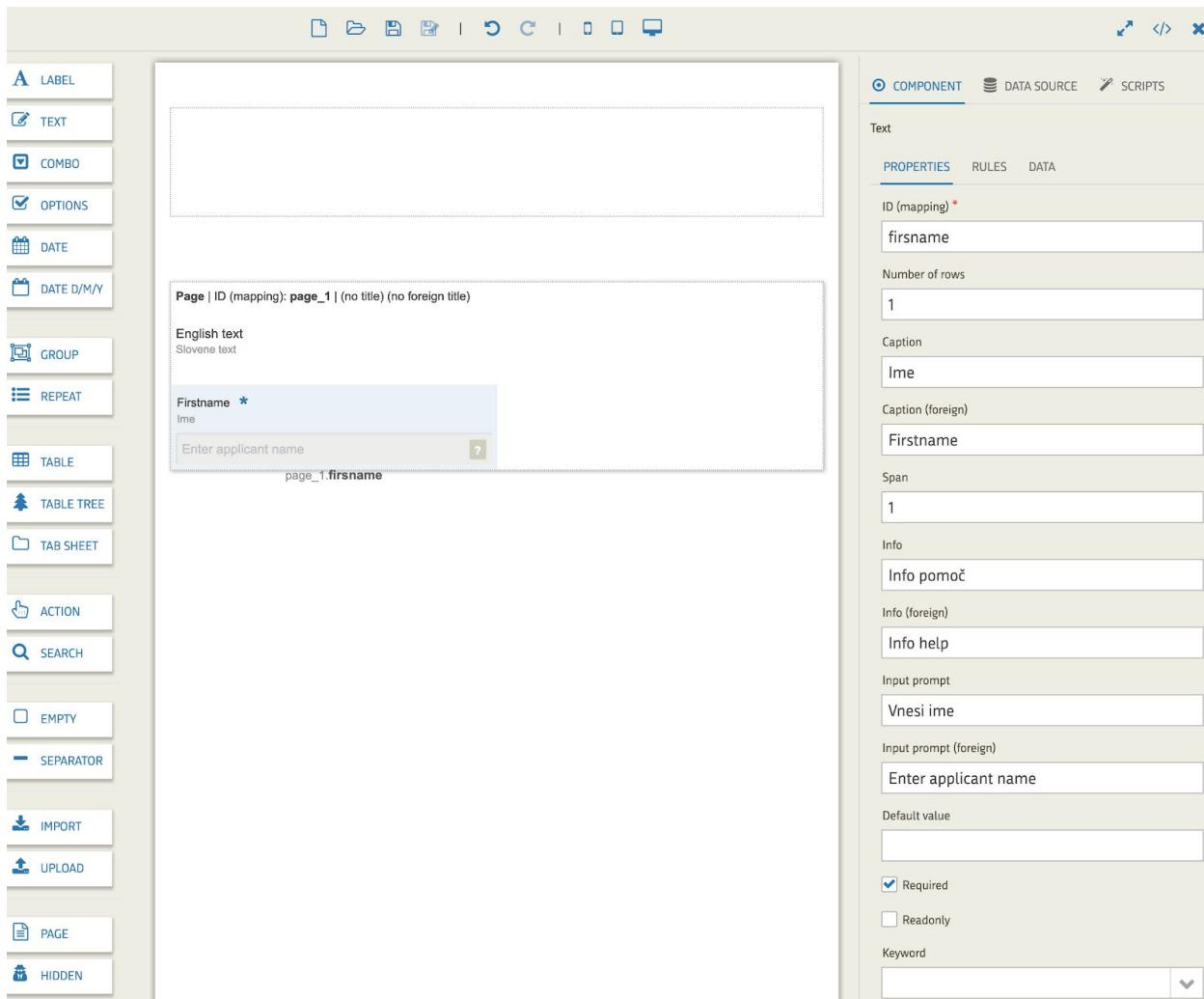
Picture 4.15 - Labels

Labels can have any HTML values and styles. Use XML editor to add any specific HTML styling. It can be also used for mapping information from the external data sources.

```
<label mapping="label_2" value="Please enter user name and password" span="3" separator="false"/>
```

#### 4.3.2.2. Text

Text is one of the most important components that enables data entry such as names, surnames, addresses, etc. The size of the field is configurable by changing the number of lines property. It is also used for mapping data from the external data sources.

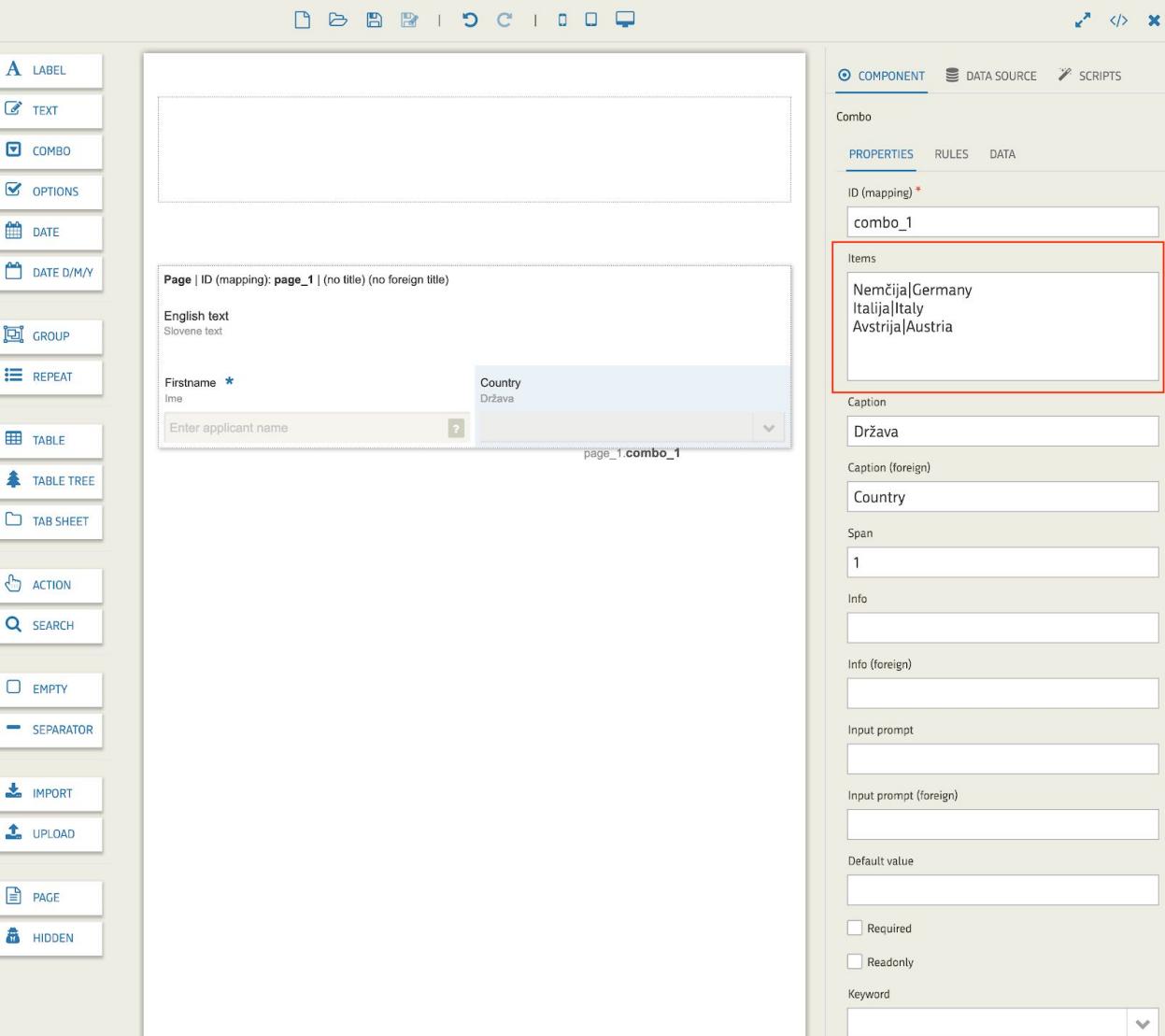


Picture 4.16 - Text field

The properties are self explanatory, except for the property “Keyword” which is beyond the scope of this documentation and is used to map this component to other nodes of the process definition.

#### 4.3.2.3. 4.3.2.3.Combo

Combo component allows the applicant to choose an item from a predefined set of options, presented in the form of a drop down menu that also supports quick filtering. It is mostly used for representing choices, for example, nationality, education, ...



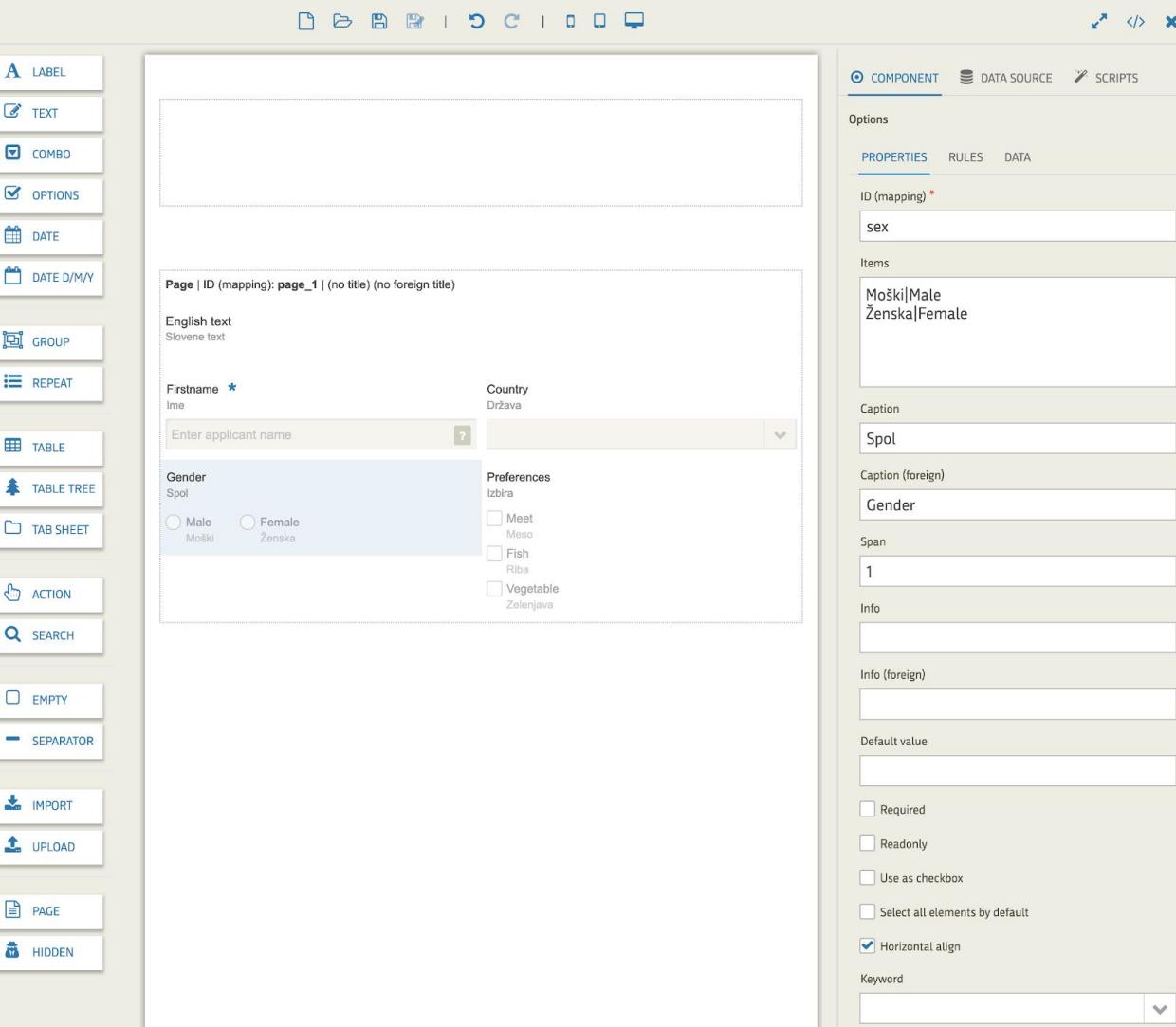
The screenshot shows the MEDIUS application interface. On the left, there is a toolbar with various components: LABEL, TEXT, COMBO (which is selected), OPTIONS, DATE, DATE D/M/Y, GROUP, REPEAT, TABLE, TABLE TREE, TAB SHEET, ACTION, SEARCH, EMPTY, SEPARATOR, IMPORT, UPLOAD, PAGE, and HIDDEN. In the center, there is a preview area showing a page with a search bar labeled "Enter applicant name" and a dropdown menu labeled "page\_1.combo\_1". The dropdown menu has three items: "Nemčija|Germany", "Italija|Italy", and "Avstrija|Austria". On the right, there is a configuration panel for the "Combo" component. The "ITEMS" section contains the list of items: "Nemčija|Germany", "Italija|Italy", and "Avstria|Austria". The "CAPTION" field is set to "Država". The "COUNTRY" field is set to "Country". The "SPAN" field is set to "1". The "INFO" field is empty. The "INFO (FOREIGN)" field is empty. The "INPUT PROMPT" field is empty. The "INPUT PROMPT (FOREIGN)" field is empty. The "DEFAULT VALUE" field is empty. The "REQUIRED" checkbox is checked. The "READONLY" checkbox is unchecked. The "KEYWORD" field is empty.

Picture 4.17 - Combo

The values for the Combo can be entered by the administrator or by defining the external data source (see chapter “Data Sources”). Combo can display values in one of two defined languages. Therefore, administrator must enter both values (e.i. Italija | Italy). On the front form, the values are displayed in the language of the portal.

#### 4.3.2.4. Options

The Options component allows the user to choose one of a predefined set of options. The options can either be presented as radio buttons (one option to be selected), or as checkboxes (multi selection). A common and powerful use for the Options component is triggering rules and/or scripts based on the users choice. For example, hiding or showing a group component, if the user selects “yes” or “no”, etc. (see chapter “Rules”).



The screenshot shows the MEDIUS application interface. On the left is a toolbar with various component icons: Label, Text, Combo, Options (selected), Date, Date D/M/Y, Group, Repeat, Table, Table Tree, Tab Sheet, Action, Search, Empty, Separator, Import, Upload, Page, and Hidden. The main area is divided into two sections: a preview window on the left and a properties editor on the right.

**Preview Window:**

- Shows a form field labeled "Page | ID (mapping): page\_1 | (no title) (no foreign title)".
- Contains an "English text" and "Slovene text" section.
- A "Firstname \* Ime" input field with placeholder "Enter applicant name".
- A "Country Država" dropdown menu.
- A "Gender Spol" section with radio buttons for "Male Moški" and "Female Ženska".
- A "Preferences Izbira" section with checkboxes for "Meet Meso", "Fish Riba", and "Vegetable Zelenjava".

**Properties Editor (Options Component):**

- Component:** Options
- Properties:**
  - ID (mapping) \*: SEX
  - Items: Moški|Male, Ženska|Female
  - Caption: Spol
  - Caption (foreign): Gender
  - Span: 1
  - Info: (empty)
  - Info (foreign): (empty)
  - Default value: (empty)
  - Required: (unchecked)
  - Readonly: (unchecked)
  - Use as checkbox: (unchecked)
  - Select all elements by default: (unchecked)
  - Horizontal align: (checked)
  - Keyword: (empty)

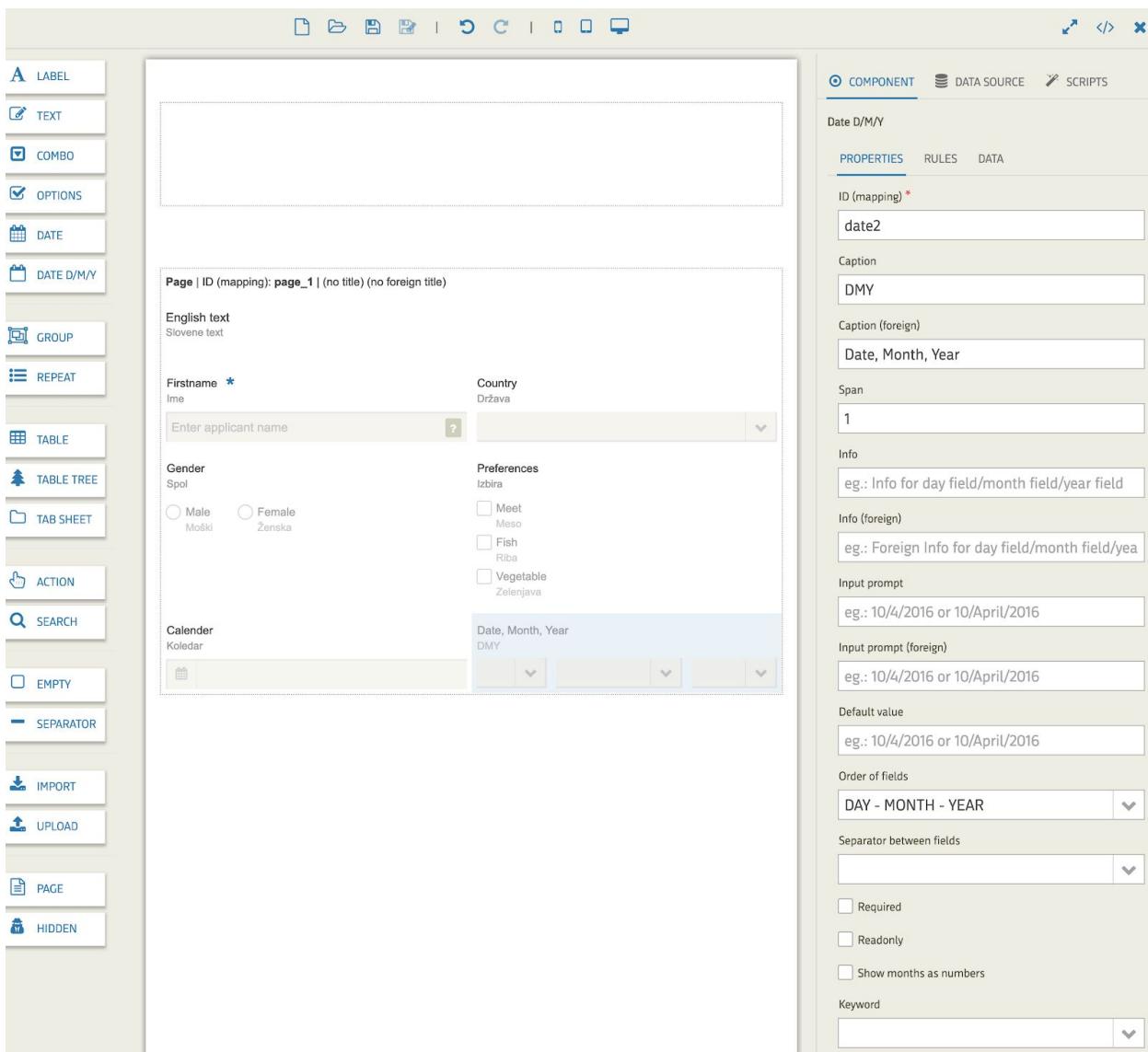
Picture 4.18 - Options can be used for both Radio and Check buttons

The property values of the Options components are self explanatory.

#### 4.3.2.5. Date Components

Editor enables two types of date components. The Date component is used for quick and easy date selection, using a popup visual calendar, that opens when the calendar icon on the field is clicked.

The Date D/M/Y component is used for date selection with the separate fields for day, month and year. The field order and the separator are both configurable. It is used for choosing dates in a specific order, or with a specific separator.

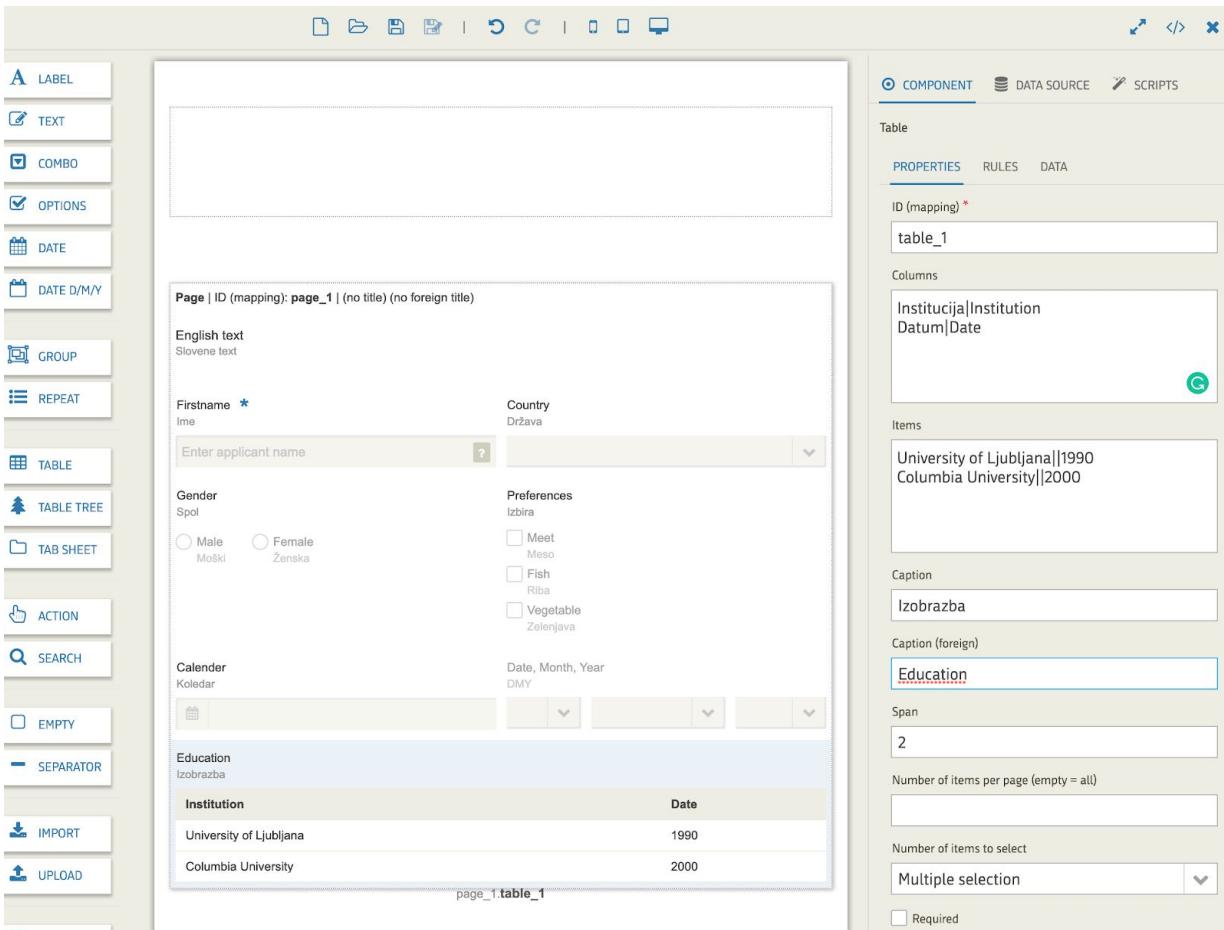


Picture 4.19 - Two types of date components

The property values are self explanatory. The property “Order of fields” determines date format, while the property “Separator” defines three separators (slash, dot, dash).

#### 4.3.2.6. Table

This component displays data in a table. The number of columns and their sizes are all configurable. By default, the component is read only, but can also be configured to enable single or multiple row selection. Used for displaying data in a table, and optionally, enabling single or multiple row selection. The values for Table can be entered by the administrator or by gathering data from the external data source (see chapter “Data Sources”).



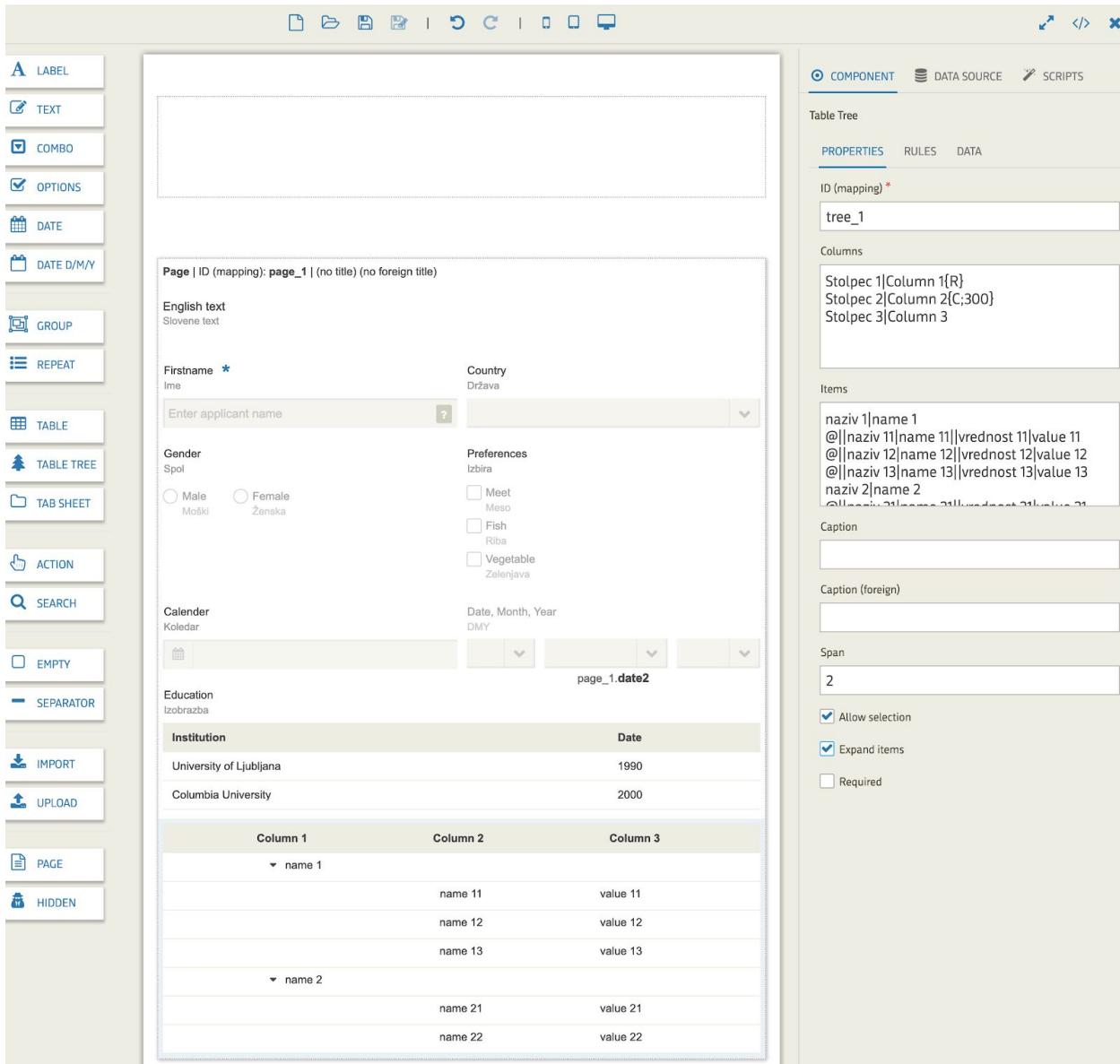
Picture 4.20 - Table component

The property values are self explanatory. On the front form, the table values are displayed only in the language of the portal.

#### 4.3.2.7. Table tree

This component displays data in a table with expandable/collapsible rows replicating the tree structure of its items. The number of columns and their sizes are all configurable. By default, the

component is read only, but can also be configured to enable single or multiple row selection. Used for displaying data in a table, where each row can have multiple sub rows, and optionally, enabling single or multiple row selection.



The screenshot shows the MEDIUS component editor interface. On the left, there is a toolbar with various component icons: Label, Text, Combo, Options, Date, Date D/M/Y, Group, Repeat, Table, Table Tree, Tab Sheet, Action, Search, Empty, Separator, Import, Upload, Page, and Hidden. The main workspace displays a form with several fields:

- Page | ID (mapping): page\_1 | (no title) (no foreign title)**
- English text**: Slovene text
- Firstname \***: Ime
- Country**: Država
- Gender**: Spol

  - Male: Moški
  - Female: Ženska

- Preferences**: Izbera

  - Meet: Meso
  - Fish: Riba
  - Vegetable: Zelenjava

- Calender**: Koledar
- Date, Month, Year**: DMY
- Education**: Izobrazba

Institution	Date
University of Ljubljana	1990
Columbia University	2000

Column 1	Column 2	Column 3
▼ name 1		
	name 11	value 11
	name 12	value 12
	name 13	value 13
▼ name 2		
	name 21	value 21
	name 22	value 22

The right panel shows the configuration for the **Table Tree** component:

- Properties** tab: ID (mapping) \* tree\_1
- Columns** section: Stolpec 1|Column 1|R, Stolpec 2|Column 2|C;300, Stolpec 3|Column 3
- Items** section: naziv 1|name 1 @||naziv 11|name 11||vrednost 11|value 11 @||naziv 12|name 12||vrednost 12|value 12 @||naziv 13|name 13||vrednost 13|value 13 naziv 2|name 2 @||naziv 21|name 21||vrednost 21|value 21
- Caption**: (empty)
- Caption (foreign)**: (empty)
- Span**: 2
- Allow selection**: checked
- Expand items**: checked
- Required**: unchecked

Picture 4.21 - TableTree component

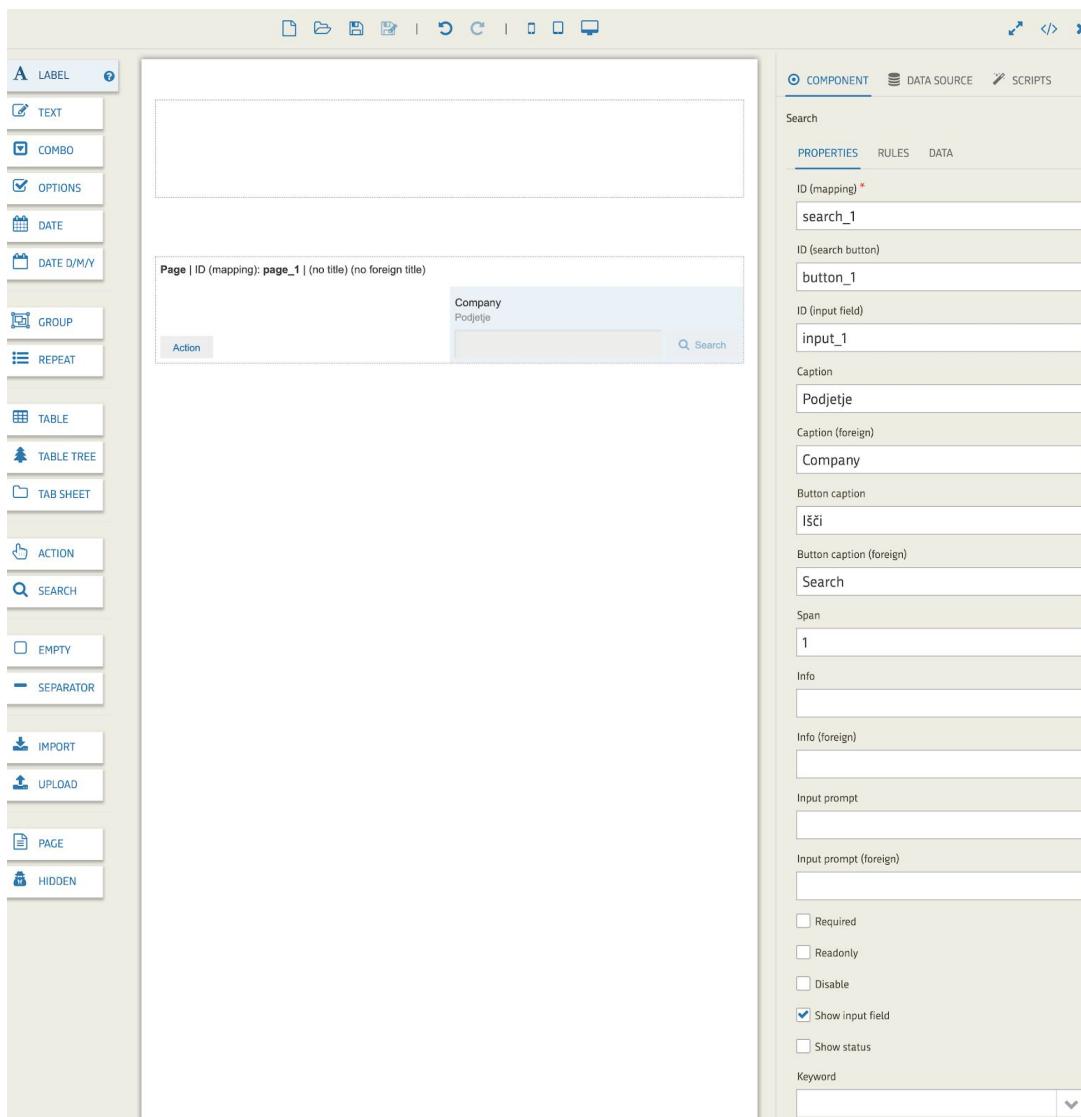
The property values are self explanatory. Use character @ to define the next level of rows. On the front form, the table values are displayed in the language of the portal.

#### 4.3.2.8. Action

A button which can be configured to trigger rules, scripts and data sources. See chapters Rules and Data Sources. The property values are self explanatory.

#### 4.3.2.9. Search

This component is a composite component of Text and Action components. It is used to trigger data gathering from the external data sources. See chapter “Data Source” for more details.



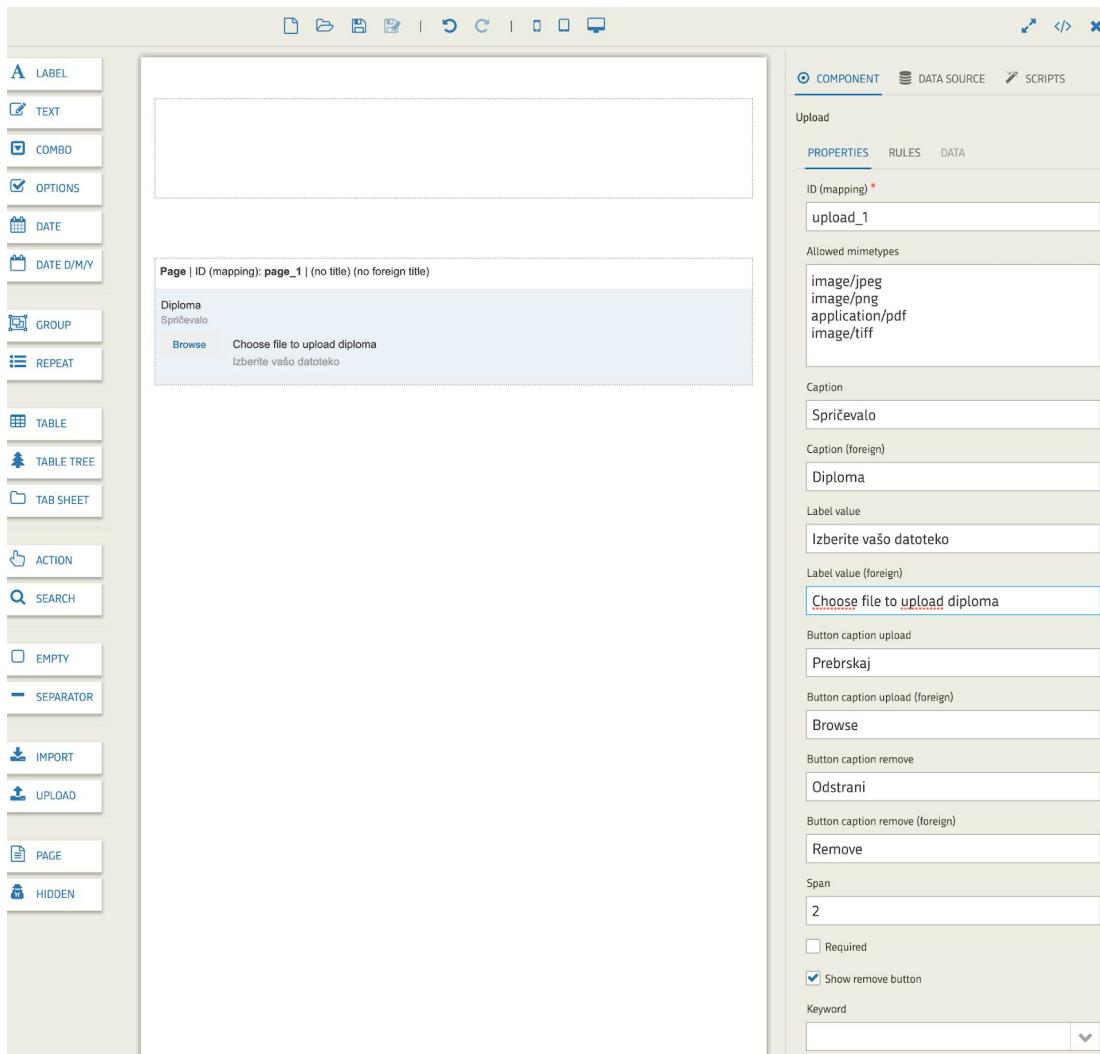
Picture 4.22 - Trigger buttons (Action and Search)

---

The property values are self explanatory.

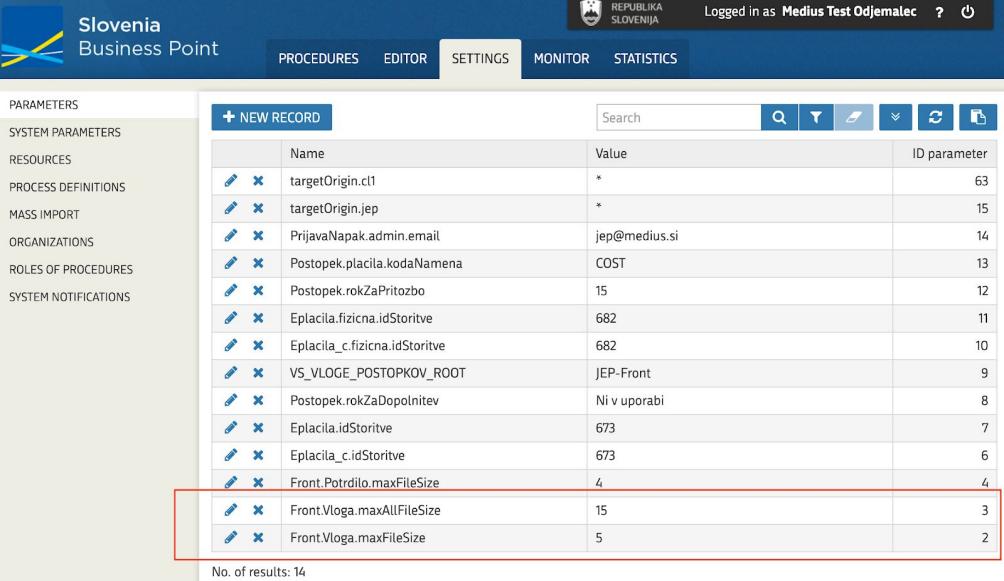
#### 4.3.2.10. Upload

This component enables uploading of files/attachments to the form by the applicant. Only files of the allowed mimetypes can be uploaded. It is used for acquiring required applicant's documents, for example, identification or other legal documents, education diploma...



Picture 4.23 - Upload component

The allowed mime types can be defined by the administrator. However, the size of the uploaded files can be customized by the system wide parameters `maxFileSize` and `maxAllFileSize`.



The screenshot shows a web-based application interface for 'Slovenia Business Point'. At the top, there's a navigation bar with tabs: PROCEDURES, EDITOR, SETTINGS (which is selected), MONITOR, and STATISTICS. On the left, a sidebar lists various system components: PARAMETERS, SYSTEM PARAMETERS, RESOURCES, PROCESS DEFINITIONS, MASS IMPORT, ORGANIZATIONS, ROLES OF PROCEDURES, and SYSTEM NOTIFICATIONS. The main content area displays a table titled '+ NEW RECORD' with columns for Name, Value, and ID parameter. The table contains 14 rows of system parameters. A red box highlights the last three rows:

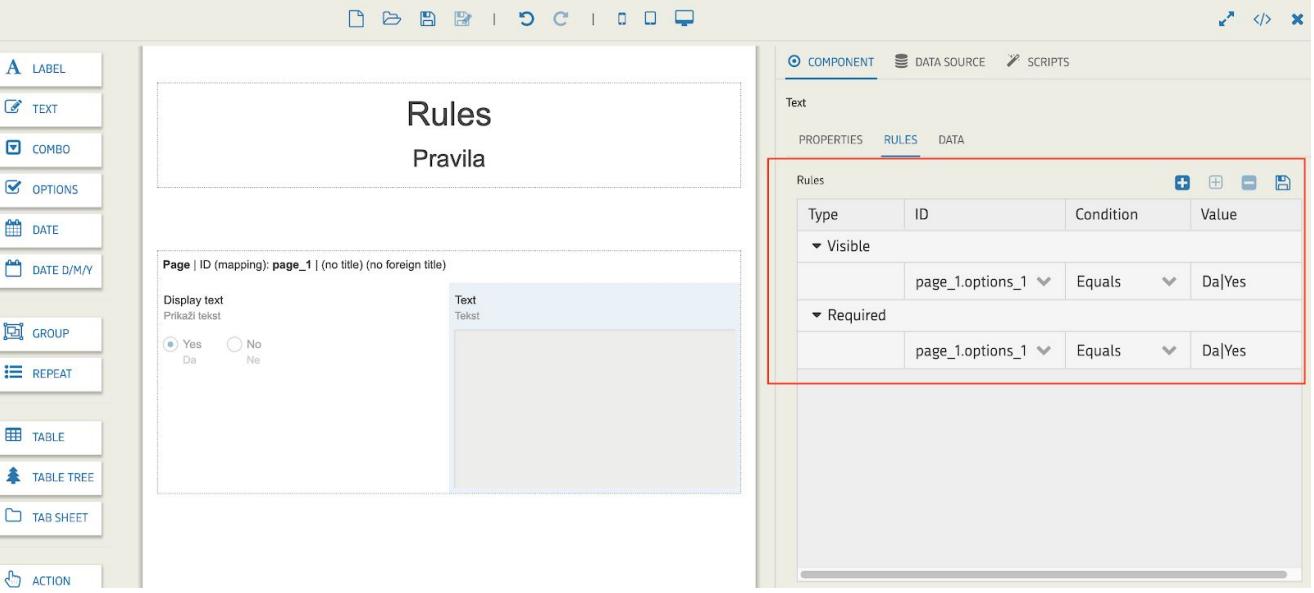
Name	Value	ID parameter
targetOrigin.cl1	*	63
targetOrigin.jep	*	15
PrijavaNapak.admin.email	jep@medius.si	14
Postopek.placila.kodaNamena	COST	13
Postopek.rokZaPrizozbo	15	12
Eplacila.fizicna.idStoritve	682	11
Eplacila_c_fizicna.idStoritve	682	10
VS_VLOGE_POSTOPKOV_ROOT	JEP-Front	9
Postopek.rokZaDopolnitve	Ni v uporabi	8
Eplacila.idStoritve	673	7
Eplacila_c_idStoritve	673	6
Front.Potrdilo.maxFileSize	4	4
Front.Vloga.maxAllFileSize	15	3
Front.Vloga.maxFileSize	5	2

No. of results: 14

Picture 4.24 - Sistem wide parameters

### 4.3.3. Rules

For each component it is possible to define a set of rules that apply to that particular component. For example, picture 4.25 shows a form with two components. The first component is a simple Yes/No radio button. The second component is a text area which is visible and required if the button Yes is pressed.



The screenshot shows a user interface for defining rules. On the left, there's a toolbar with icons for various components like Label, Text, Combo, Options, Date, etc. The main area has a title 'Rules' and a subtitle 'Pravila'. Below that, there's a section for 'Page | ID (mapping): page\_1 | (no title) (no foreign title)'. It contains a 'Display text' field with the placeholder 'Tekst' and a 'Text' field below it. To the right, there's a 'RULES' panel with tabs for PROPERTIES, RULES, and DATA. The RULES tab is selected, showing a table with columns: Type, ID, Condition, and Value. Two rows are visible under the 'Visible' section:

Type	ID	Condition	Value
Visible	page_1.options_1	Equals	Da Yes

Below that, another section for 'Required' is shown:

Required	page_1.options_1	Equals	Da Yes
----------	------------------	--------	--------

Picture 4.25 - Rule definition

---

Possible actions: *ENABLE, READONLY, VISIBLE, REQUIRED*.

Possible conditions: *Equals, Not equals, Empty, Not empty, Is valid, Is not valid, Is verified, Is not verified*.

The XML presentation of the text field having two rules looks as following:

```
<text rows="10" mapping="text_1" caption="Tekst" captionForeign="Text" span="1" required="false" readonly="false" id="962397376">
    <rules>
        <rule type="visible">
            <expression mapping="page_1.options_1" condition="equals" value="Da|Yes"/>
        </rule>
        <rule type="required">
            <expression mapping="page_1.options_1" condition="equals" value="Da|Yes"/>
        </rule>
    </rules>
</text>
```

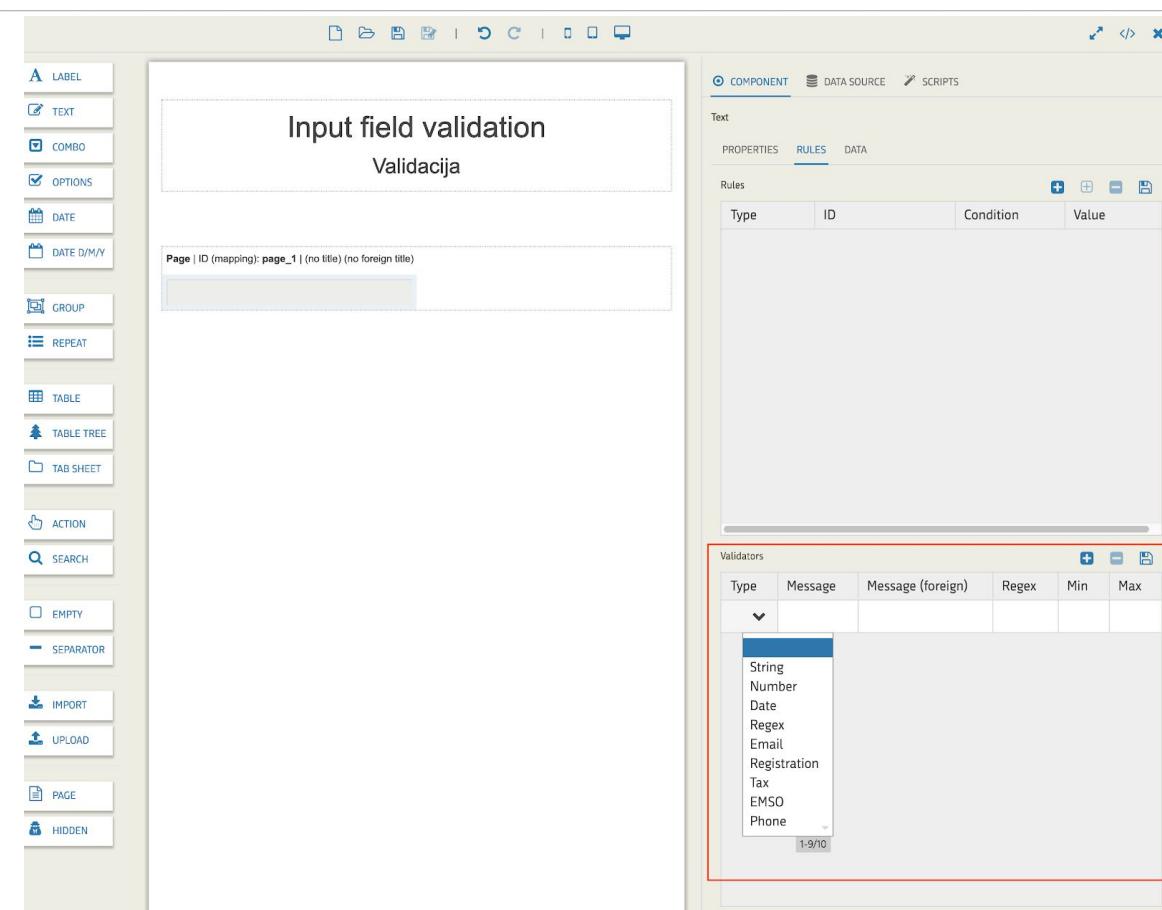
Picture 4.26 - XML presentation of the rule

#### 4.3.4. Validations

For every component of type "Text", it is possible to define the validation rules. The following is the list of supported validators:

- String - verifies the length of the input text (between "Min" and "Max").
- Number - the input text is numeric value (between "Min" and "Max"). Validation checks the integers and decimal numbers.
- Date - checks if the input text matches the date format.
- Email - Verify that the text input is in accordance with the e-mail format.
- Registration - verify that the input text matches the format for the company/person registration number.
- Tax - checks if the input text is in accordance with the tax code format.
- EMŠO - checks if the text input is in accordance with the EMŠO format
- Phone - checks if the text input is in accordance with the phone number format.
- IBAN - verifies that the input text is in accordance with the format for the IBAN number.
- Regular expression - checks if the entry is in line with the regular expression written by the administrator. The regular expression is entered in the regex field.

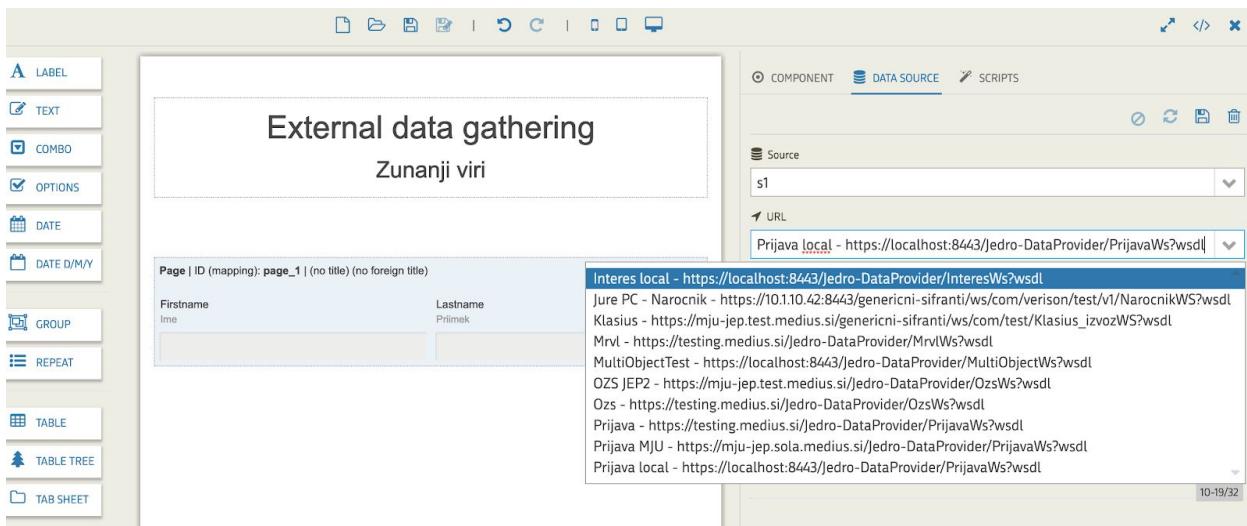
Picture 4.27 shows the process of creating a validator in the property editor of the component.



Picture 4.27 - Validation of input text fields

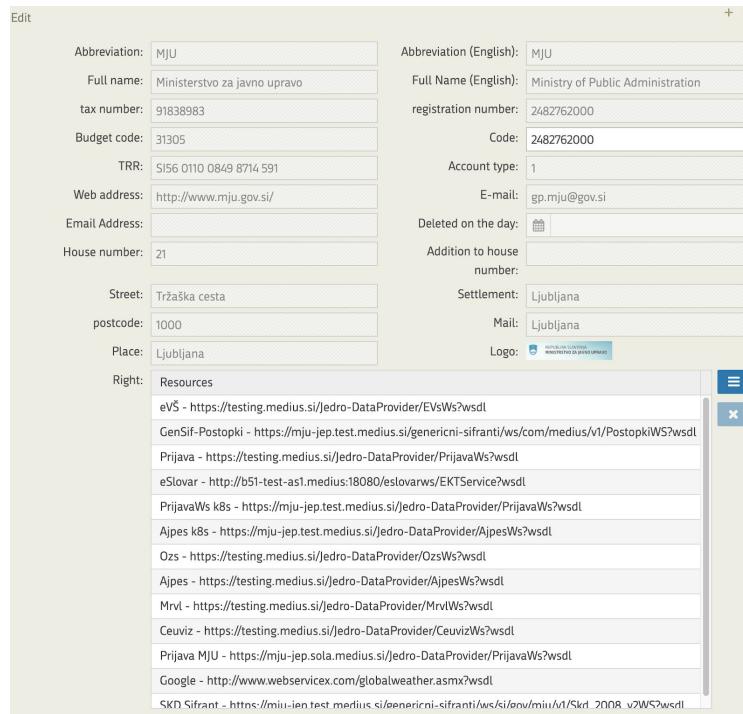
#### 4.3.5. Data Sources

Mapping data between the external data sources and the graphical components is one of the most useful functionalities of the application form editor. Data from the external sources can be mapped to labels, texts, combos, tables, tabs, repeatables, trees. The form editor is capable of gathering data from any SOAP web service and map it to the GUI component. However, only the allowed SOAP web services are available as a list of possible external data sources. Picture 4.28 shows the process of choosing a data source from the list of available data source.



Picture 4.28 - Choosing an external data source

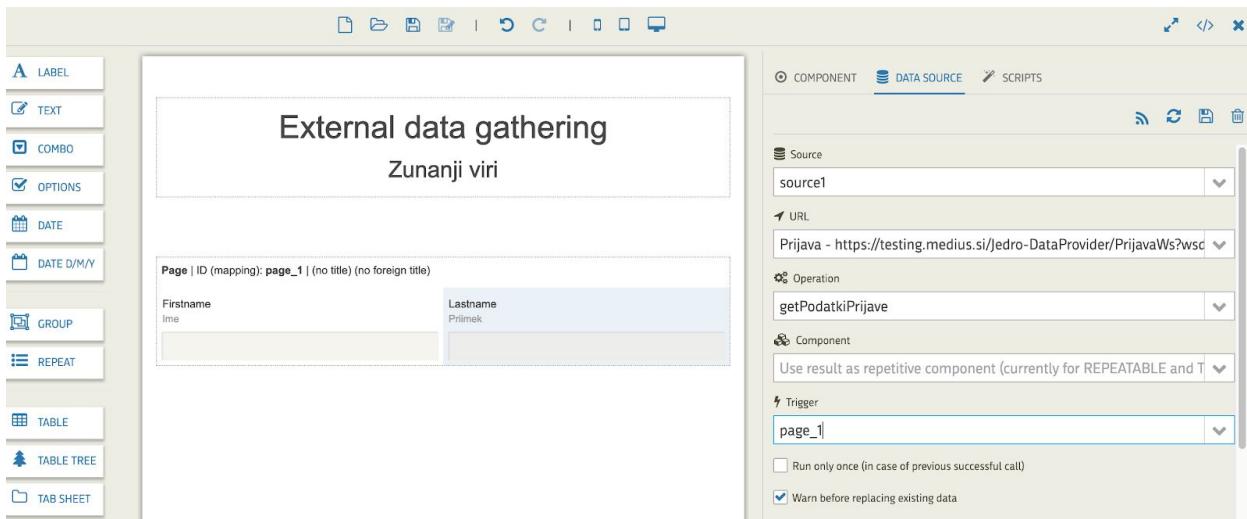
Available SOAP web services are managed by the administrator in the system administration console. Go to SETTINGS → ORGANIZATIONS and choose one of the organizations to see available SOAP services that can be used by the administrators of that organization. In general, organizations are owners of the application forms and only the administrators of the organization can design forms. Picture 4.29 shows a mask to manage external data sources for the particular organization.



Picture 4.29 - Managing external data sources

When one of the available SOAP web services is chosen (for example, *Prijava* - <https://testing.mediust.si/Jedro-DataProvider/PrijavaWs?wsdl>) the following two properties must be defined for that service:

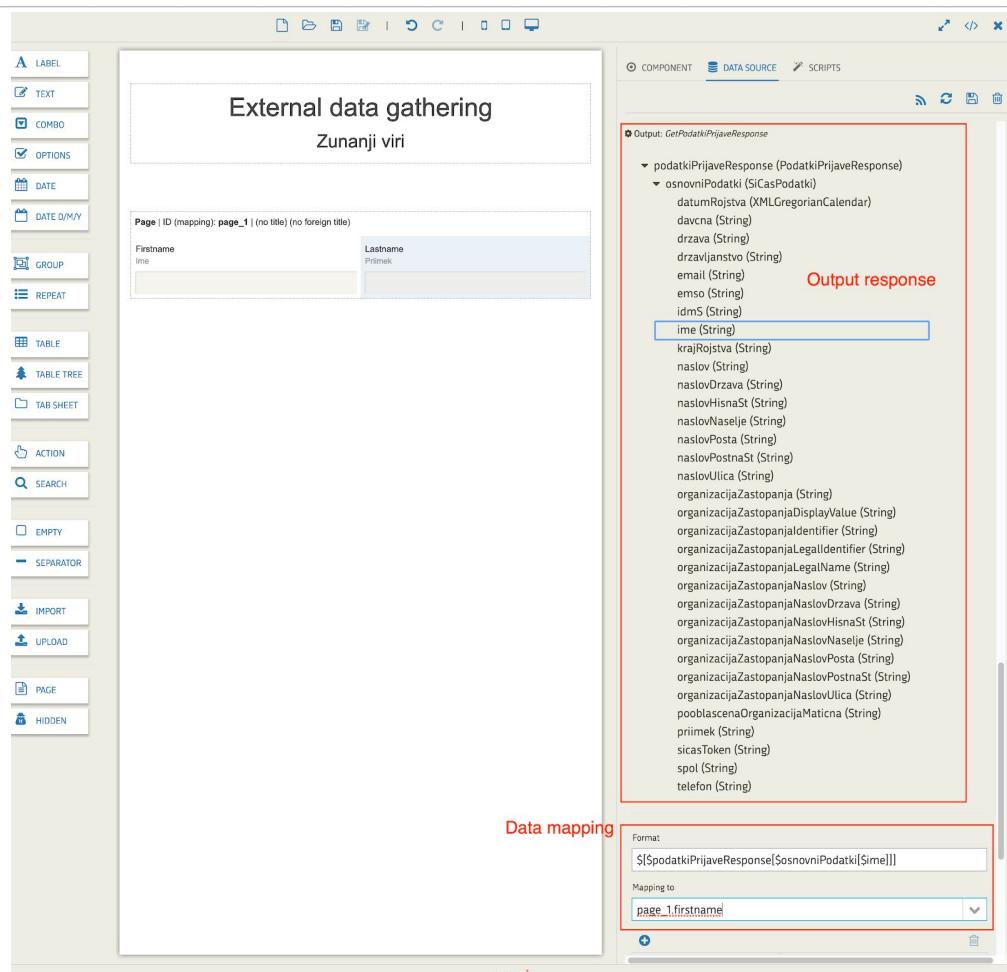
- Operation - a SOAP web service method to be called (*getPodatkiPrijeve*).
- Trigger - an id of the component that triggers data gathering. For example, picture 4.30 shows that data will be gathered when page *page\_1* is opened.



Picture 4.30 - Choosing SOAP service, method and trigger component

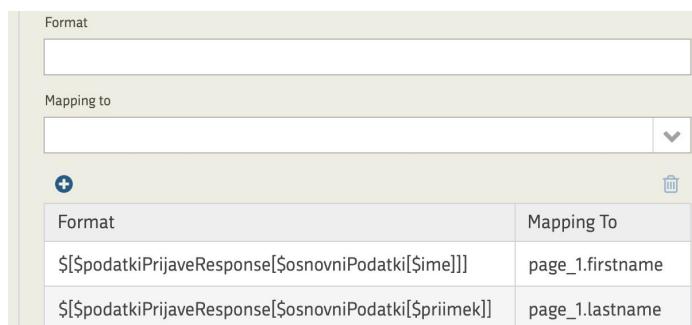
#### 4.3.5.1. Mapping data

The SOAP service <https://testing.mediust.si/Jedro-DataProvider/PrijavaWs?wsdl> returns a complex XML response structure. The output structure of the service is displayed as a tree-like structure by the editor as shown in picture 4.31. The administrator can click on one of the output fields (`$[$podatkiPrijeveResponse[$osnovniPodatki[$ime]]]`) and map it to one of the components on the canvas (`page_1.firstname`). Button  confirms the selection.



Picture 4.31 - Mapping SOAP response data to text field

Any number of output parameters and any combination of them can be mapped to the canvas component. The result of data mapping is shown in picture 4.32.



Picture 4.32 - Mapped components

Button Save  saves the configuration and administrator can test the gathering process of the WEB service. When a page is opened, two input fields (firstname, lastname) are automatically filled with data gathered from the external service.



Input Preview PDF DATA XML X

## External data gathering

### Zunanji viri

Firstname	Lastname
Ime	Priimek
Janez	Novak

Picture 4.33 - Testing gathering data from external SOAP services

#### 4.3.6.1. Repeatable data gathering

Data gathering is mostly associated with the components that display repeatable items (Table, TableTree, Combo, Tab, Repeatable). For example, mapping selected SOAP output parameter to table component is shown in the following picture.

Format	Mapping To
<code>[\$vozilo[\$mpVin    \$znamka\n]]</code>	page_1.vehicles

Two bars `||` are used to separate values in the columns; `\n` is used to define a new line. The resulting table visualization is as in picture 4.34. Table displays VIN (Vehicle Identification Number) and Brand of cars gathered from the official MRVL registry. Applicant can select one or more items from the table.

## External data gathering

### Zunanji viri

Vehicles  
Vozila

Vehicle Identification Number	Brand
WV2ZZZ2KZ6X085288	VOLKSWAGEN
WV2ZZZ2KZ8X015657	VOLKSWAGEN
<b>VF77JBHZMFJ769235</b>	CITROEN
VF77JBHZMFJ771296	CITROEN
<b>WV2ZZZ2KZ7X109535</b>	VOLKSWAGEN
VF77JBHZMGJ809196	CITROEN
<b>VF77JBHZMGJ809197</b>	CITROEN
<b>WV2ZZZ2KZ7X050508</b>	VOLKSWAGEN
VF3GJ9HWC95232737	PEUGEOT

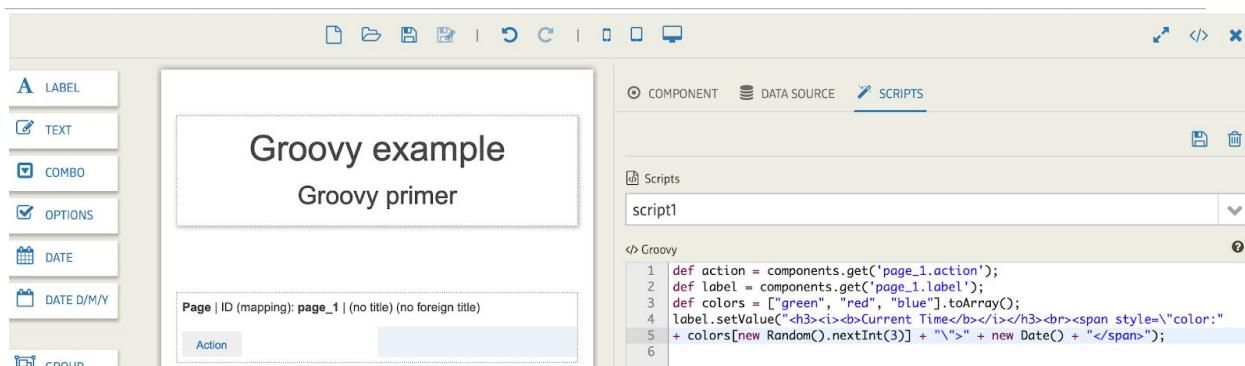
Picture 4.34 - table item selection gathered from the external vehicle registry

### 4.3.7. Groovy Script

There are application forms that require functionalities and form behaviour not provided by default by the existing components. Most of that can be overtaken by the specific programming logic administrators can write as Groovy scripts. In the upper right corner there is button "SCRIPTS" which opens a Groovy script editor. Each canvas component can be represented as a variable in a Groovy script. For example, let assume a simple form as in picture 4.35. There are two components on the page. The first one is an action button, the second component is a blank label. These two components are represented in a Groovy script as:

```
def action = components.get('page_1.action');
def label = components.get('page_1.label');
```

Every time an applicant clicks on the button, a label displays current time colored with a random color.



Picture 4.35 - Simple Groovy script

For a sample of a Groovy script functionality check <https://demo.mediusti/MediustForms> and click button Groovy.

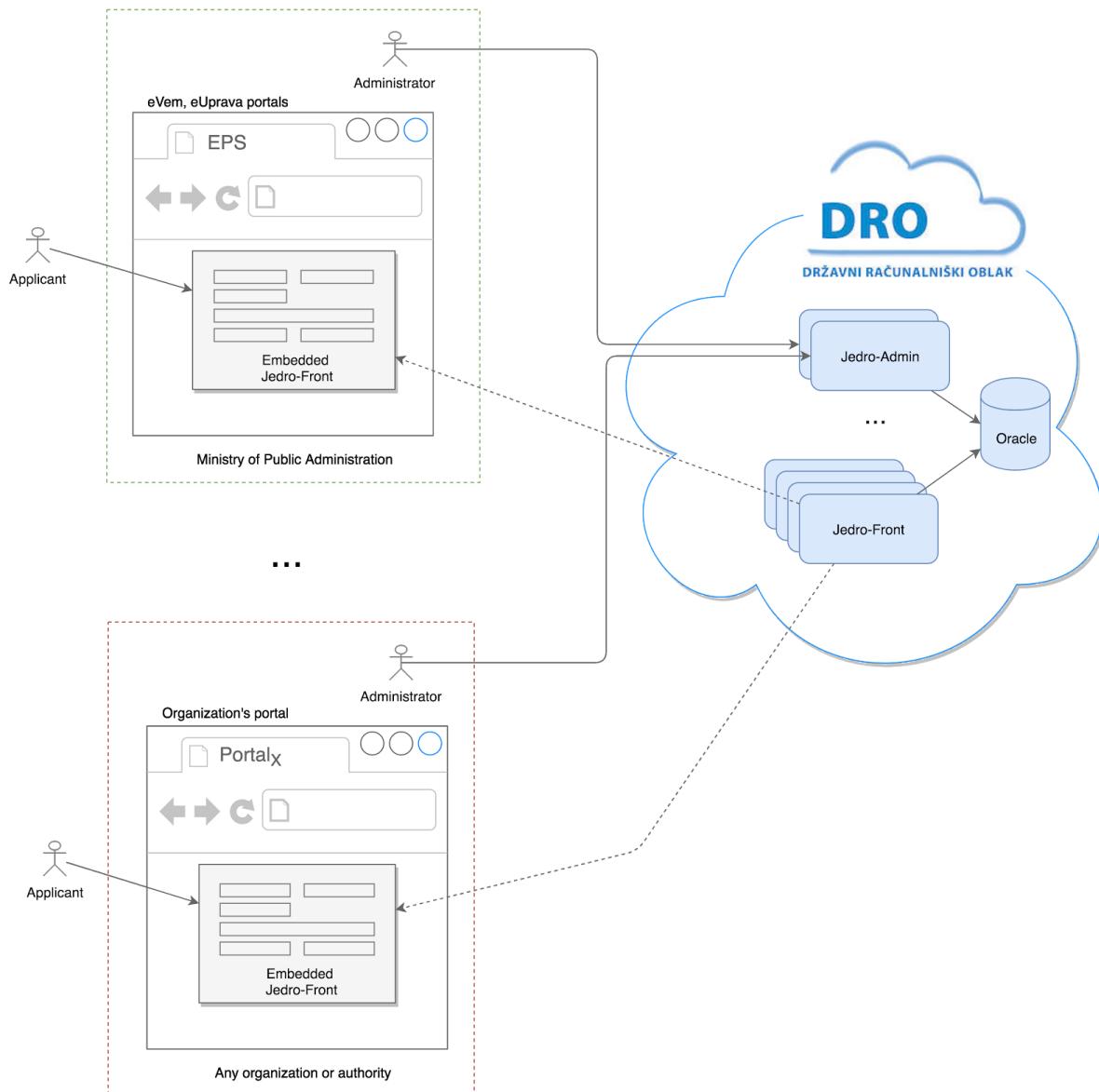
## 5. Software as a Service

IS EPS is based on the most innovative cloud based technology running as SaaS (Software as a Service) type system in the private cloud called DRO - Državni Računalniški Oblak (Slovene State Computer Cloud). It is in production since 2016. IS EPS includes 12 self managed Docker container microservices. Microservice images are managed and orchestrated by the Kubernetes which provides real-time upscaling and downscaling of microservices, load balancing, automatic rollout of services, etc. The following is a list of some of the most important microservices:

- Jedro-admin - tool for managing procedures, process definitions, application forms and other system properties.
- Jedro-front - Process State Engine that instantiate and rendes EPS procedures. Exposes Javascript API for CORS communication.
- Jedro-DataProvider - set of web services that can be accessed as data sources by the application forms.
- Jedro-Diagnostics - monitoring and health checking of the whole system.
- Jedro-EDV - temporary persistent storage of procedures with the thorough auditing capabilities.
- Jedro-Procesnik - a simple information system for competent organizations not having their own information systems.
- Jedro-front-frame - a simple portal that embeds Jedro-front using CORS.
- ...

In this cloud based architecture, microservices can be accessed by any authorized 3rd party information systems in a way shown in picture 5.1. For example, the Ministry of Public Administration, the owner of IS EPS runs these microservices in order to publish procedures for their eVem, eUprava and JEP portals. Moreover, any organization with the proper

authentication and authorization privileges can use these services the same way as the Ministry of Public Administration does. The only thing these organizations must do is to make their portals embed Jedro-front. This is possible by using the provided CORS JavaScript API to exchange data between Jedro-front and the HTML pages of the portal. As far as the administration is concerned, everything is the same. Administrators use the same Jedro-admin service, but with the access rights restricted to their organization's procedures only.



Picture 5.1 - SaaS architecture of IS EPS

EPS microservices are self managed and therefore can be used for multi-institutional purposes, nationwide, even cross-border. Microservices need not to run in DRO. They are build to run in any private or public cloud.

## 5.1. Technology

Microservice Jedro-front is developed by Vaadin.

The rest of the microservices are developed by JEE for server-side functionality (Wildfly, Oracle) and MediusCommons for front end functionality. MediusCommons is an open-source tool for rapid WEB based application development based on Vaadin and JEE. It is developed by Medius.

Microservice Jedro-front-frame is developed by JEE for server-side functionality (Wildfly, Oracle) and SpringMVC/JEE for front end functionality.

Microservices are build by Thorntail and packed into container images using Docker.

Kubernetes is used for microservice orchestration in the cloud. Kubernetes is the best open-source container orchestration system.

## 6. MediusFormsDemo

MediusFormsDemo is a simple HTML application that demonstrates the functionality of the application form editor, without the complexity of Jedro-admin where process definitions, procedures, process state actions, access rights, competent organizations, external data source, etc. must be managed prior to starting creating the application form. MediusFormsDemo does not require any authentication or authorisation and can therefore be used by any Internet user.

MediusFormsDemo is available at URL: <https://demo.mediustech.com/MediusForms/>

The application form editor at MediusFormsDemo lacks some functionalities compared to the application form editor in Jedro-admin of IS EPS. For example, a new SOAP web service can not be configured by the editor. This is obvious, since the access rights for the SOAP web services are managed outside the application form editor. However, users can test and try all other functionalities, such as creating form layouts, define input components, build rules and validations, write Groovy scripts, previews, etc. Some predefined forms are available and it is strongly suggested to open and try at least the following sample forms:

- SAMPLE FIELDS - application form demonstrates how to create input components (labels, texts, dates, combo boxes, options, tables, trees, tabs, uploads, ...).
- LAYOUT - application form demonstrates most of the layout techniques using pages, groups, tabs and repeatables.

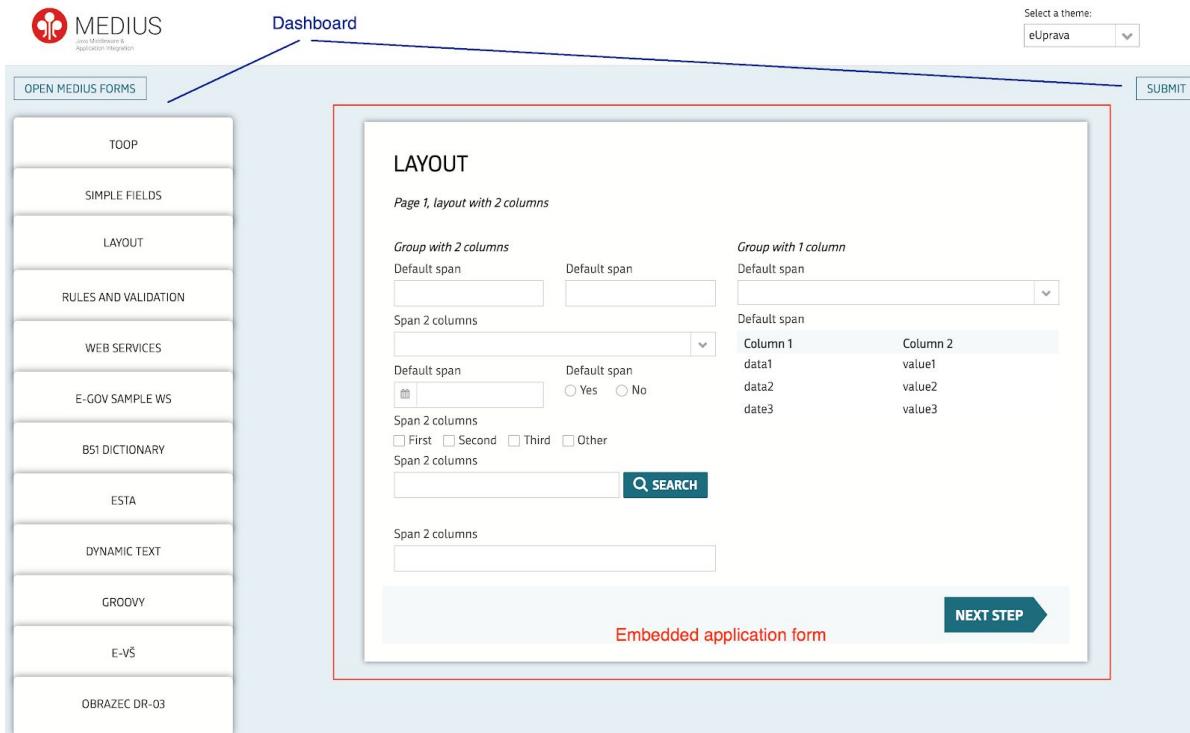
- RULES AND VALIDATIONS - application form demonstrates use of rules and validations to manage components behaviour.
- GROOVY - application form demonstrates use of Groovy scripts to manage components behaviour.

The rest of the available application form samples are very specific for some known usecases and they are not interesting for the general public use.

The landing page of the MediusFormsDemo is a simple dashboard as shown in picture 6.1. On the left side of the dashboard is a list of buttons, each button starts a form. Every time a new form is created the new button appears on the list. In order not to spam the demo with a wild number of forms (buttons), the MediusFormsDemo is reinitialized every night and all forms are

deleted. In the upper right corner of the dashboard, there is a combo-box of possible displaying themes . Three themes are available: eUprava (the default, look and feel of the official government portal), Compact (look & feel of the EPS portal Jedro-front-frame), Cozy (look & feel of [www.medijs.si](http://www.medijs.si)).

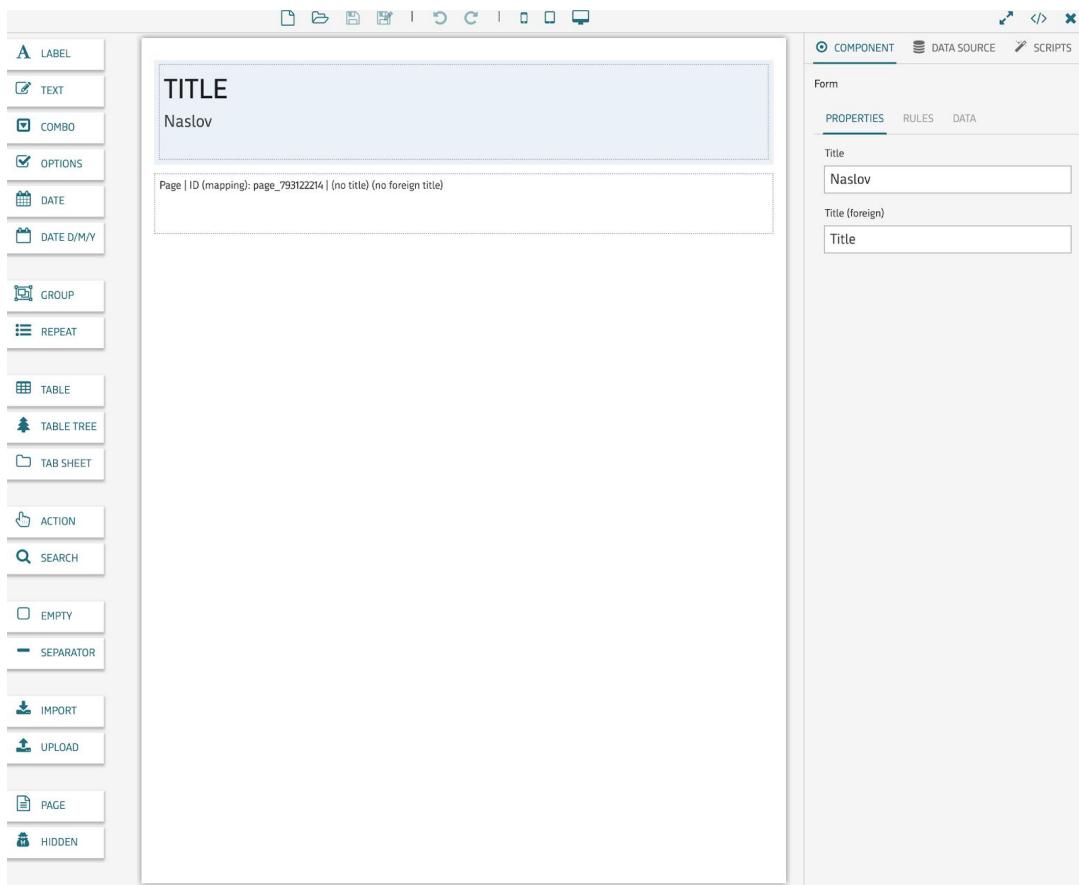
The inner part of the dashboard is a running application form. Users can fill data the same way they would do it in the application form running in the Jedro-front of IS EPS. Button SUBMIT is a fake simulation of a form submission. It just displays the content of the form (PDF view and XML data).



The screenshot shows the MediusFormsDemo landing page. At the top, there's a header with the MEDIUS logo and contact information. Below the header is a navigation menu on the left with various buttons: OPEN MEDIJS FORMS, TOOP, SIMPLE FIELDS, LAYOUT, RULES AND VALIDATION, WEB SERVICES, E-GOV SAMPLE WS, BSI DICTIONARY, ESTA, DYNAMIC TEXT, GROOVY, E-VŠ, and OBRAZEC DR-03. A blue arrow points from the word "Dashboard" to the "LAYOUT" button in the menu. To the right of the menu, there's a large application form window titled "LAYOUT". The form contains several sections: "Group with 2 columns" (with two input fields), "Group with 1 column" (with one input field), "Default span" (with three rows: "Column 1" containing "data1", "data2", "data3" and "Column 2" containing "value1", "value2", "value3"), and "Span 2 columns" (with a dropdown menu showing "First", "Second", "Third", and "Other"). There are also "SEARCH" and "NEXT STEP" buttons at the bottom of the form. In the top right corner of the dashboard area, there's a "Select a theme:" dropdown set to "eUprava".

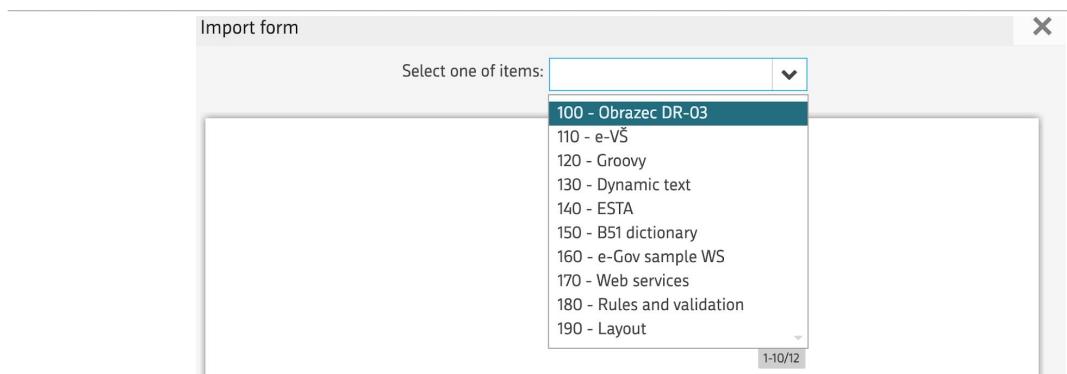
Picture 6.1 - MediusFormsDemo landing page

The most important button on the dashboard is **OPEN MEDIUS FORMS** in the left upper corner. It opens the application form editor the same way as it is opened in the Jedro-admin when the administrator clicks the “FORM EDITOR” button in the menu EDITOR → FORM EDITOR, or when the administrator clicks the feature button of the process definition action “Obrazec”. A new and empty application form is displayed on the canvas.



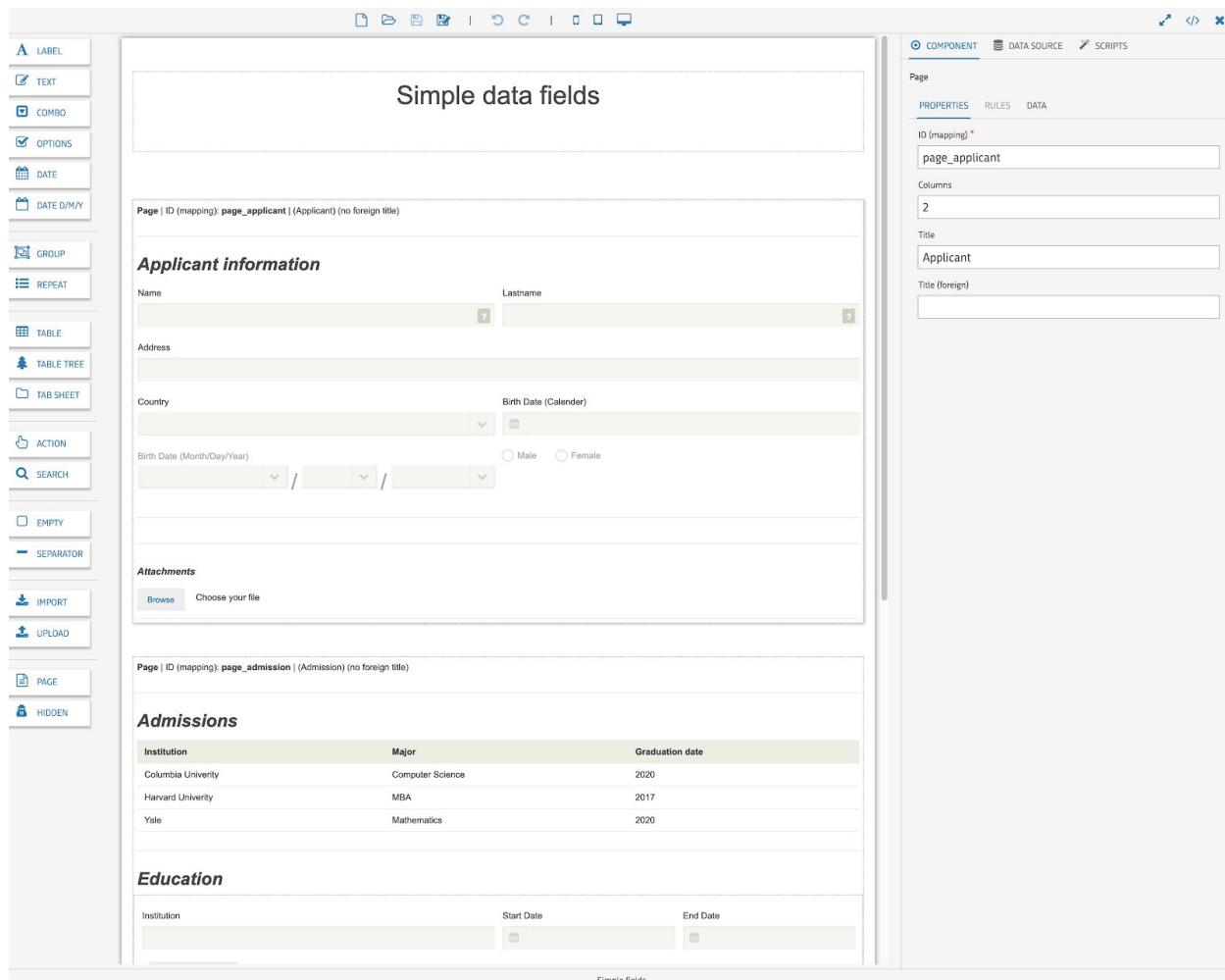
Picture 6.2 - Application form editor

In order to select one of the sample forms from the list of the application forms the button  is used. The provided combo box list the names of all available application forms.



Picture 6.3 - Selecting a sample form

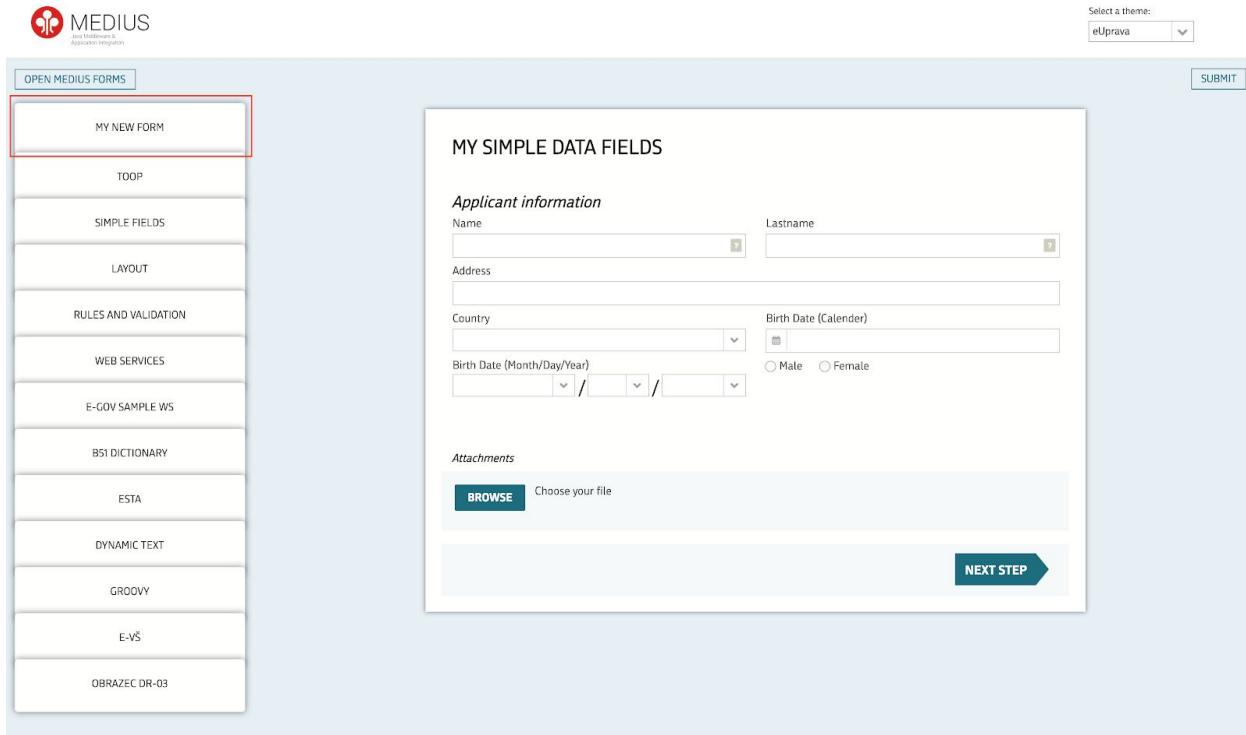
By selecting one of the available forms and clicking the OK button   , the form appears in the editor.



Picture 6.3 - Sample application form

You can modify any component in the editor or start a new form. Since the application forms are represented as XML files, it is also possible to create a new form by just copy/pasting one of the XML forms created before.

Button “Save As”  creates a new application form that appears in the list of available forms.



The screenshot shows the MEDIUS application interface. On the left, there is a sidebar with various options: OPEN MEDIUS FORMS, TOOP, SIMPLE FIELDS, LAYOUT, RULES AND VALIDATION, WEB SERVICES, E-GOV SAMPLE WS, B51 DICTIONARY, ESTA, DYNAMIC TEXT, GROOVY, E-VS, and OBRAZEC DR-03. The 'MY NEW FORM' option is highlighted with a red border. To the right, a main panel displays a form titled 'MY SIMPLE DATA FIELDS'. This form includes sections for 'Applicant information' (Name, Lastname, Address, Country, Birth Date (Calender), Birth Date (Month/Day/Year), Male/Female radio buttons) and 'Attachments' (BROWSE button, Choose your file input field). At the bottom right of the main panel is a 'NEXT STEP' button with a right-pointing arrow. At the top right of the main panel, there is a dropdown menu labeled 'Select a theme:' with 'eUprrava' selected, and a 'SUBMIT' button.

Picture 6.4 - A new form appears in the dashboard

## 7. Video demos

Two videos are available to demonstrate simple IS EPS use.

Jedro-admin: [https://dev.mediust.si/files/EPS\\_Admin.mp4](https://dev.mediust.si/files/EPS_Admin.mp4)

Jedro-front-frame: [https://dev.mediust.si/files/EPS\\_Front.mp4](https://dev.mediust.si/files/EPS_Front.mp4)

The following application XML form is used in the video demo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<form title="" titleForeign="" id="32160864" xmlns="http://www.mediust.si/mediusforms/schema/Form">
<pages>
  <page mapping="page_applicant" title="Vlagatelj" titleForeign="Applicant" id="1209713678">
```

```

<layout columns="2">
    <separator value="Podatki o vlagatelju" valueForeign="Applicant data" span="2" id="2006267809"/>
        <text rows="1" mapping="firstname" caption="Ime" captionForeign="Firstname" span="1" required="false"
    readonly="true" id="1159777163"/>
            <text rows="1" mapping="lastname" caption="Priimek" captionForeign="Lastname" span="1" required="false"
    readonly="true" id="1152181761"/>
                <text rows="1" mapping="address" caption="Naslov" captionForeign="Address and house number" span="2"
required="true" readonly="false" id="796247045"/>
                    <text rows="1" mapping="phone" caption="Telefon" captionForeign="Phone" span="1" required="false"
readonly="false" id="1714006302">
                        <rules>
                            <rule type="validate">
                                <validator type="phone"/>
                            </rule>
                        </rules>
                    </text>
                    <text rows="1" mapping="mail" caption="Elektronska pošta" captionForeign="E-mail" span="1" required="true"
readonly="false" id="1222799962">
                        <rules>
                            <rule type="validate">
                                <validator type="email" message="Neveljaven email naslov" messageForeign="Not a legal email address"/>
                            </rule>
                        </rules>
                    </text>
                    <dateDMY mapping="date" caption="Datum rojstva" captionForeign="Date of Birth" span="1" order="DMY"
required="false" readonly="false" monthAsNumber="false" id="470983086"/>
                    <combo mapping="country" caption="Država rojstva" captionForeign="Country of Birth" span="1" required="false"
readonly="false" id="1037708688"/>
                </layout>
            </page>
<page mapping="page_work_permit" title="Dovoljenje" titleForeign="Work permit" id="1321411541">
    <layout columns="2">
        <separator value="Podatki o zaposlitvi" valueForeign="Work permit data" span="2" id="2046728348"/>
        <tableTree mapping="classification" caption="Klasifikacija" captionForeign="Work classification" span="2"
allowSelection="true" expand="false" required="true" id="765507356">
            <columns value="Naziv|Activit name"/>
        </tableTree>
        <br span="2" id="75999401"/>
        <label mapping="label" value="Izobrazba" valueForeign="Education" span="2" separator="true" id="862747119"/>
        <repeatable mapping="education" span="1000" buttonCaptionAdd="Dodaj nov zapis" buttonCaptionAddForeign="Add
new " buttonCaptionRemove="Odstrani zapis" buttonCaptionRemoveForeign="Remove" hideAddButton="false"
hideRemoveButton="false" showAddIcon="false" showRemoveIcon="false" id="586993103">
            <layout columns="2">
                <text rows="1" mapping="institution" caption="Naziv" captionForeign="Institution name" span="2" required="false"
readonly="false" id="1182542352"/>
                <combo mapping="edu_type" caption="Izberite ustrezno vrsto izobrazbe" captionForeign="Choose appropriate
education" span="1" required="true" readonly="false" id="2021460780">
                    <items value="Nepopolna nižja stopnja osnovnošolske | Incomplete higher level of primary education"/>
                    <items value="Srednja splošna | Secondary general"/>
                    <items value="Visokošolska strokovna | Professional education "/>
                    <items value="Magisterij znanosti | Master's degree "/>
                    <items value="Doktor znanosti | Doctoral degree "/>
                </combo>
                <text rows="1" mapping="date" caption="Datum" captionForeign="Date of completion" span="1" required="false"
readonly="false" id="797155646"/>
            </layout>
        </repeatable>
    </layout>
</page>
</pages>
<sources>
    <source name="Internal source" noRepetitive="false" internal="true"/>
    <source name="classification" service="provided:SKD Sifrant" operation="getAll" trigger="page_work_permit">

```

```

noRepetitive="true" warn="false" internal="false">
    <output mappingTo="page_work_permit.classification" format="$[${skd2008V2Tabela[$angleskiDeskriptor \n
$skd2008V2Tabela2[@$angleskiDeskriptor\n]]}]">
    </source>
    <source name="login" service="provided:Prijava" operation="getPodatkiPrijava" trigger="page_applicant"
noRepetitive="false" warn="false" internal="false">
        <input property="application.applicationIdentifier" type="java.lang.String"/>
        <input property="mode.operation" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.datumRojstva" type="javax.xml.datatype.XMLGregorianCalendar"/>
        <input property="vlagatelj.osnovniPodatki.davcna" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.drzava" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.drzavljanstvo" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.email" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.emso" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.idmS" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.ime" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.krajRojstva" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naslov" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naslovDrzava" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naslovHisnaSt" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naselje" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naslovPosta" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naslovPostnaSt" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.naslovUlica" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanja" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaDisplayValue" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjalIdentifier" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaLegalIdentifier" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaLegalName" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslov" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslovDrzava" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslovHisnaSt" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslovNaselje" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslovPosta" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslovPostnaSt" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.organizacijaZastopanjaNaslovUlica" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.pooblašcenaOrganizacijaMaticna" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.priimek" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.sicasToken" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.spol" type="java.lang.String"/>
        <input property="vlagatelj.osnovniPodatki.telefon" type="java.lang.String"/>
        <output mappingTo="page_applicant.firstname" format="$[${podatkiPrijavaResponse[$osnovniPodatki[$ime]]}]">
        <output mappingTo="page_applicant.lastname" format="$[${podatkiPrijavaResponse[$osnovniPodatki[$priimek]]}]">
    </source>
    <source name="country" service="provided:Države" operation="getAll" trigger="page_applicant" noRepetitive="true"
warn="false" internal="false">
        <output mappingTo="page_applicant.country" format="$[${sifrantDrzav[$uradniKratkiNazivDrzave\n]}]">
    </source>
</sources>
<scripts/>
</form>

```