Sii Source Code Pattern Generator

WebIDE Extension

Mateusz Skadorwa

1. Table of content

1. Table of content 1
2. Protocol 1
3. About the product 2
4. Architecture of solution 3
4.1 How to read source code? 3
4.2 How to find cursor? 4
5. How to extend WebIDE 4
Case 1. You just have source code, and you would like to extend Web IDE 4
Case 2. You use a Web IDE Extension from SAP App Center 5

2. Protocol

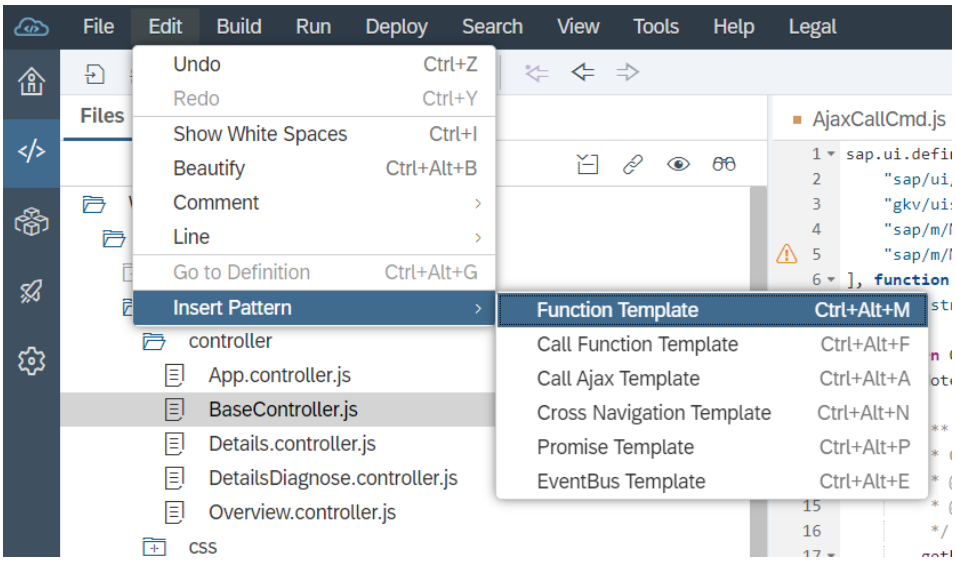
Date	Responsible for	Change
20.04.2020	Mateusz Skadorwa	Initial document

3. About the product

Sii Source Code Pattern Generator is a WebIDE Extension. It enables developers to accelerate the development due to the prepared templates of different objects like:

- Promises
- EventBus
- Functions
- Call Functions (BOPF Action)
- Cross Application Navigation
- Ajax Call

The product provides a quality of development, because the source codes are in accordance with SAP UI5 specification and JSDoc. How does it work? Extension reads the currently opened tab in WebIDE Editor, it looks for the place where your cursor stays and it puts there templates. From now, you haven't to take care about semantic. You haven't to look for the statements in your old projects or in community. Templates are gathered in one common place or you can explore them over shortcuts.



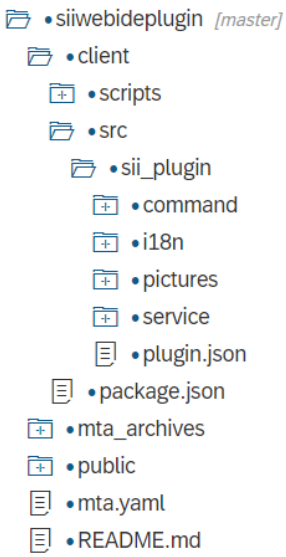
Picture 1 - Source code pattern generator. Functionality: Add Function Template

```
20
21      /**
22      * @summary
23      * @description
24      * @public
25      * @param vInput  {type} var description
26      * @return {type} Return value description
27      */
28      functionName: function (vInputParam){
29          var vPar1 = vInput;
30      return vPar1;
31      }
32
```

Picture 2 - Result of action 'Add Function Template'

4. Architecture of solution

Source code pattern generator uses the WebIDE SDK, which allows to create SAP WebIDE Extensions. The structure of extension looks like this:



The `client` folder also contains a `package.json` file, which describes the extension and its metadata. There are two important folders (`command`, `service`) and file (`plugin.json`). Folder `Command` contains all services, which are responsible for registering commands, triggered by user interactions and can fulfill certain tasks, such as saving the content of the currently shown editor. Folder `Service` contains all functions, which are responsible for interfacing with command layer and provides functionality like reading data from workspace and so on.

The `plugin.json` file contains the plugin definition, it is built as a simple JSON object, which contains plugin class name, dependencies, services, Inherent service configurations for the plugin.

More about SAP WebIDE SDK you can find in following website: https://sdk-sapwebide.dispatcher.hana.ondemand.com

4.1 How to read source code?

Module `Content` delivers the method to take care about the source code of currently opened tab. The method `getCurrentDocument` returns a promise, which give an access to tab. The method `getContent` returns the source code as a string.

```
getSourceCode: function(vSelectedRow) {
    return this.context.service.content.getCurrentDocument().then(function(oDocument) {
        return oContentAct =  oDocument.getContent().then(function(oContent){
            aCodeTab = oContent.split(/\r?\n/);
            return aCodeTab;
        });
    });
},
```

Picture 3 - how to read source code of opened tab

4.2 How to find cursor?

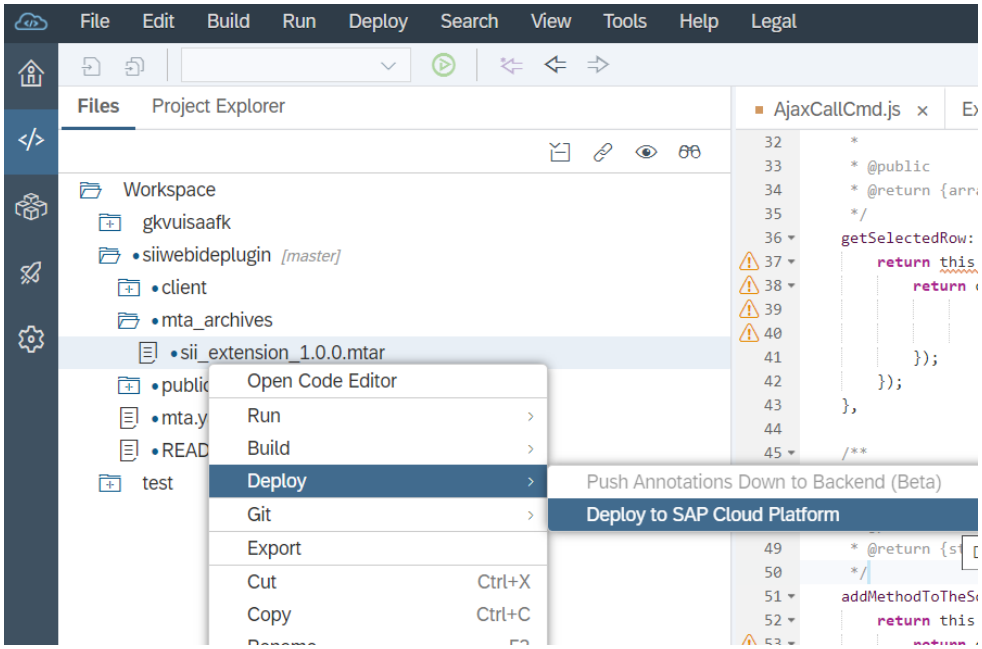
Module content, which help with the content of source code, provides the methods to read the position of cursor. Method `getFocusElement` returns the handler to get position over the method `getMarkers`.

```
getSelectedRow: function() {
    return this.context.service.content.getFocusElement().then(function(oFocus) {
        return oFocus.getMarkers().then(function(oMarker){
                vSelRow = oMarker[1].range.start.row;
                return vSelRow;
        });
    });
},
```

5. How to extend WebIDE

Case 1. You just have source code, and you would like to extend Web IDE.

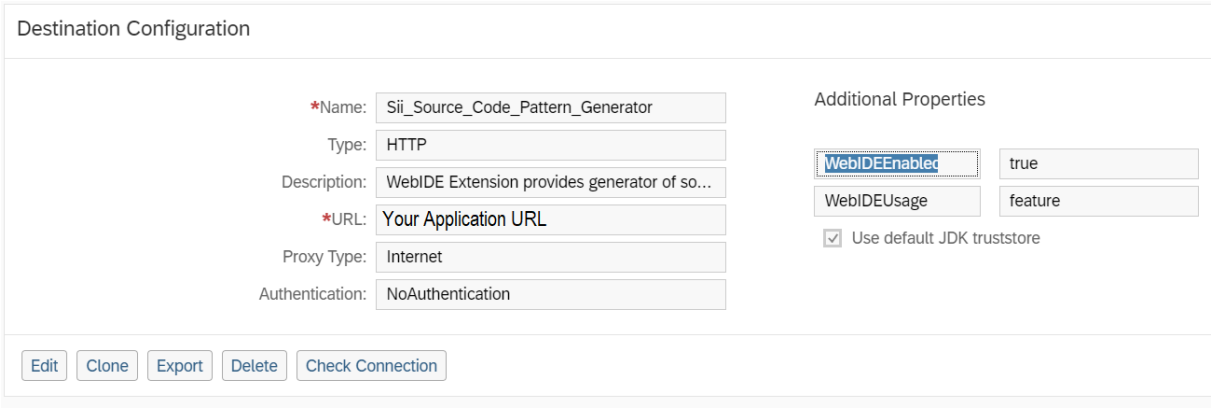
In first step, the application has to be build. Then there will be created a folder, named `mta_archives`. When it will be available, then you have to deploy build file.



Picture 4 - Deploy an application

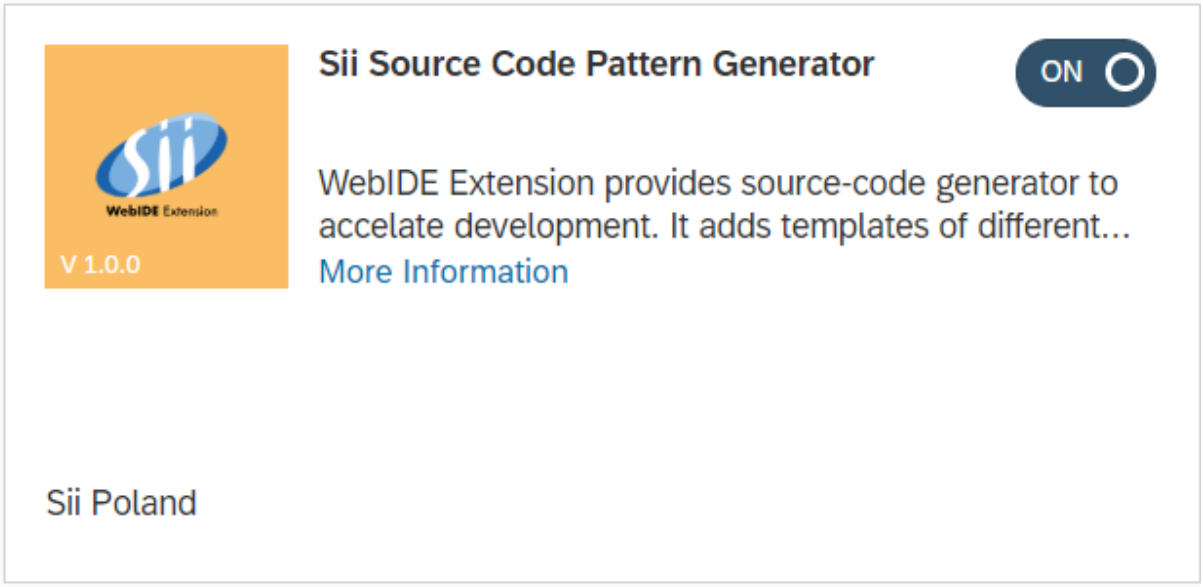
If it is done, you have to login into your SAP Cloud Platform Cockpit, and check what is the URL to the deployed application. You can find it, if you will chose your `space`, and then you have to open the `Application` tab. There you will see the deployed application. In the details of particular one, you will see the URL.

In next step, you need to configure new `Destination` in SAP Cloud Platform Cockpit. It is available under the tab `Connectivity`. The configuration of destination should looks like on the picture 5.



Picture 5 - Configuration of new destination

In final step, you have to enabled the extension in your WebIDE. In the `Preferences` tab, you have to open `Extensions`, and you need to look for Sii Source Code Pattern Generator and active the extension.



Picture 6 - Source Code Pattern Generator

Case 2. You use a Web IDE Extension from SAP App Center

You need to configure new `Destination` in SAP Cloud Platform Cockpit. It is available under the tab `Connectivity`.

The configuration should looks like at the Picture 5, but instead of “Your URL”, please put following url: https://siicf-siiprod-codegenerator-space-siiwebideplugin.cfapps.eu10.hana.ondemand.com