

```
● user: projects $ uis help
Usage: uis <command> [options]

Commands:
  uis add [--development] [--optional] <framework-libraries...>  Add SAPUI5/OpenUI5 framework libraries to the project configuration.
  uis build                                                         Build project in current directory
  uis config                                                       Get and set UIS Tooling configuration options
  uis init                                                         Initialize the UIS Tooling configuration for an application or library project.
  uis remove <framework-libraries...>                             Remove SAPUI5/OpenUI5 framework libraries from the project configuration.
  uis serve                                                         Start a web server for the current project
  uis tree                                                         Outputs the dependency tree of the current project to stdout. It takes all relevant parameters of uis build into account.
                                                                    [aliases: ls, list]
  uis use <framework-info>                                         Initialize or update the project's framework configuration.
  uis versions                                                      Shows the versions of all UIS Tooling modules

Options:
  -h, --help                Show help [boolean]
  -v, --version              Show version number [boolean]
  -c, --config               Path to project configuration file in YAML format [string]
  --dependency-definition    Path to a YAML file containing the project's dependency tree. This option will disable resolution of node package dependencies. [string]
  --workspace-config         Path to workspace configuration file in YAML format [string]
  -w, --workspace            Name of the workspace configuration to use [string] [default: "default"]
  --loglevel, --log-level    Set the logging level [string] [choices: "silent", "error", "warn", "info", "perf", "verbose", "silly"] [default: "info"]
  --verbose                  Enable verbose logging. [boolean] [default: false]
  --perf                     Enable performance measurements and related logging. [boolean] [default: false]
  --silent                   Disable all log output. [boolean] [default: false]

Examples:
  uis <command> --dependency-definition /path/to/projectDependencies.yaml  Execute command using a static dependency tree instead of resolving node package dependencies
  uis <command> --config /path/to/uis.yaml                                Execute command using a project configuration from custom path
  uis <command> --workspace dolphin                                       Execute command using the 'dolphin' workspace of a uis-workspace.yaml
  uis <command> --log-level silly                                         Execute command with the maximum log output

○ user: projects $
```

```
● user: projects $ mkdir project1
● user: projects $ cd project1
○ user: project1 $ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (project1) 
```

EXPLORER

PROJECTS

project1

webapp

index.html

package.json

package-lock.json

Welcome

Settings

Welcome

Create Intelligent Data Applications



Start

OUTPUT

DEBUG CONSOLE

TERMINAL

SQL CONSOLE

user: projects \$ cd project1

user: project1 \$ mkdir webapp

user: project1 \$ cd webapp

user: webapp \$ touch index.html

user: webapp \$

user: projects \$ cd project1

user: project1 \$ npm init

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields

and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and

save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (project1)

version: (1.0.0)

description:

entry point: (index.js)

test command:

git repository:

keywords:

author:

license: (ISC)

About to write to /home/user/projects/project1/package.json:

{

"name": "project1",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo `Error: no test specified` && exit 1"

},

"author": "",

"license": "ISC"

}

Is this OK? (yes)

user: project1 \$

←

→

Welcome

Settings

Welcome

index.html X

project1 > webapp > index.html > html

1

<!DOCTYPE html>

2

<html>

3

<head>

4

<meta charset="utf-8">

5

<title>This is webpage</title>

6

</head>

7

<body>

8

<div>

9

This is body of webpage

10

</div>

11

</body>

12

</html>

EXPLORER

...

WelcomeSettingsindex.html


PROJECTS

project1

webapp

index.htmlmanifest.jsonpackage.jsonpackage-lock.json

Create Intelligent Data Application



Start

- Create SAP HANA Database Project From Template
- Create SAP HANA Database Module From Template

Development

- Getting Started
- Create SAP HANA Database Artifact
- Start Database Artifact Recovery
- Start Support Mode Export
- Create an HDI container service instance

OUTPUTDEBUG CONSOLETERMINALSQL CONSOLE

TERMINAL

user: projects \$ cd project1

user: project1 \$ mkdir webapp

user: project1 \$ cd webapp

user: webapp \$ touch index.html

user: webapp \$ ^C

user: webapp \$ touch manifest.json

WelcomeSettingsindex.htmlmanifest.json 4 X

project1 > webapp > {} manifest.json > _version

1 {

2 "_version": "1.2.0",

3 "sap.app": {

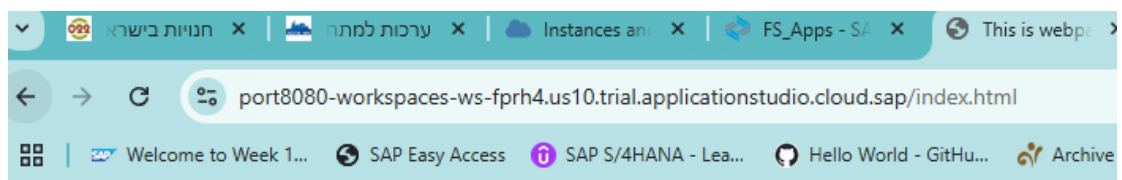
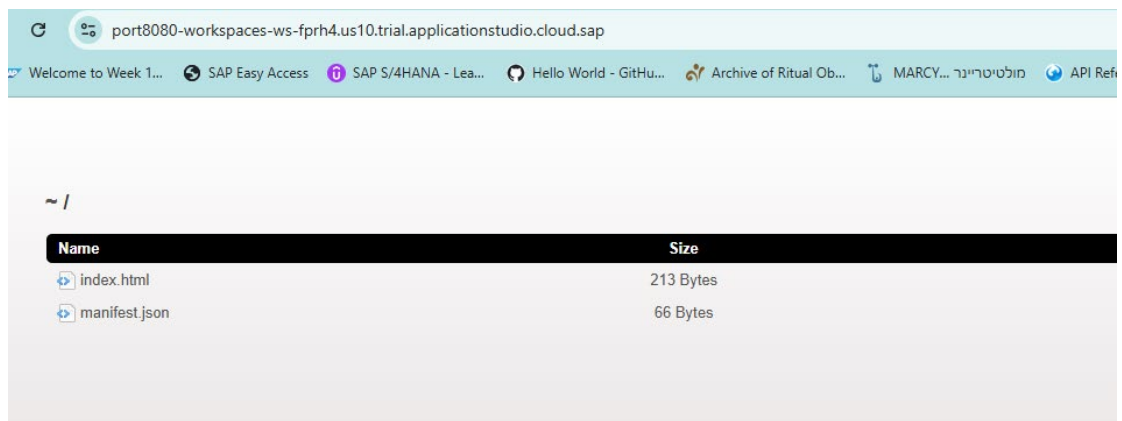
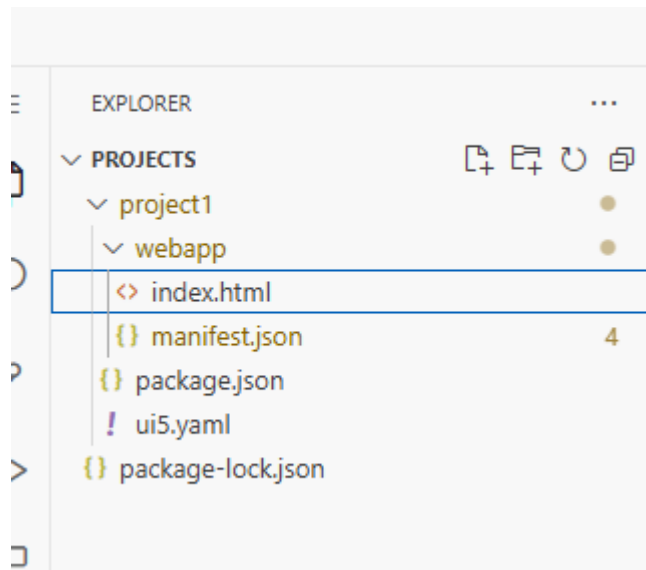
4 "id": "example.id"

5 }

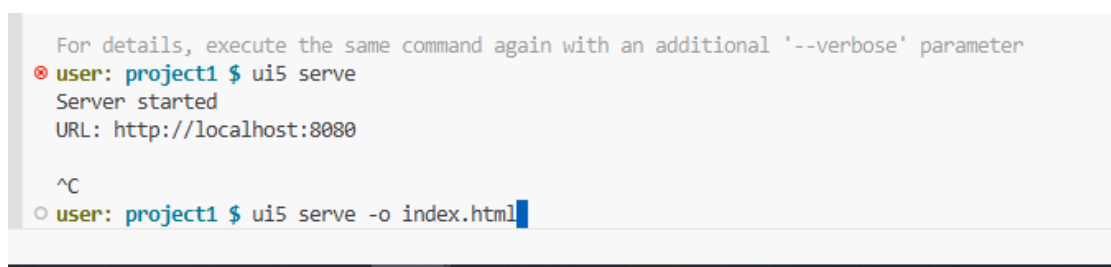
6 }

For details, execute the same command again with an additional '--verbose' parameter

- user: project1 \$ ui5 serve
Server started
URL: http://localhost:8080



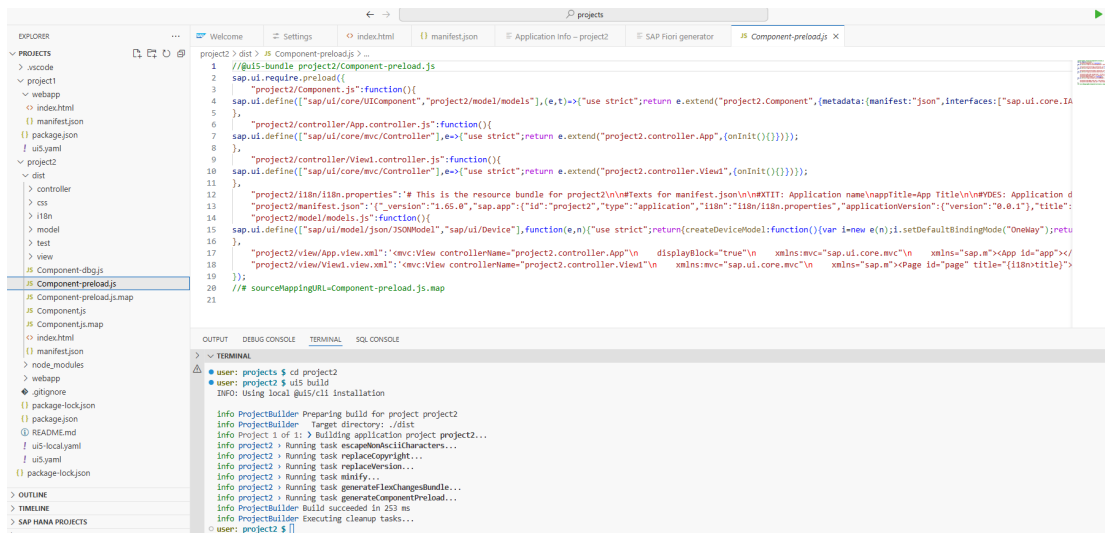
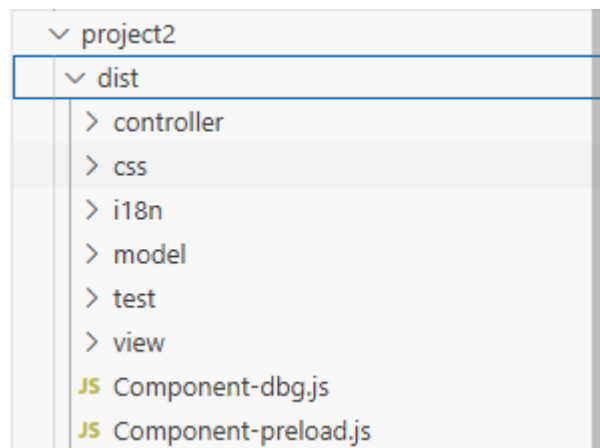
This is body of webpage

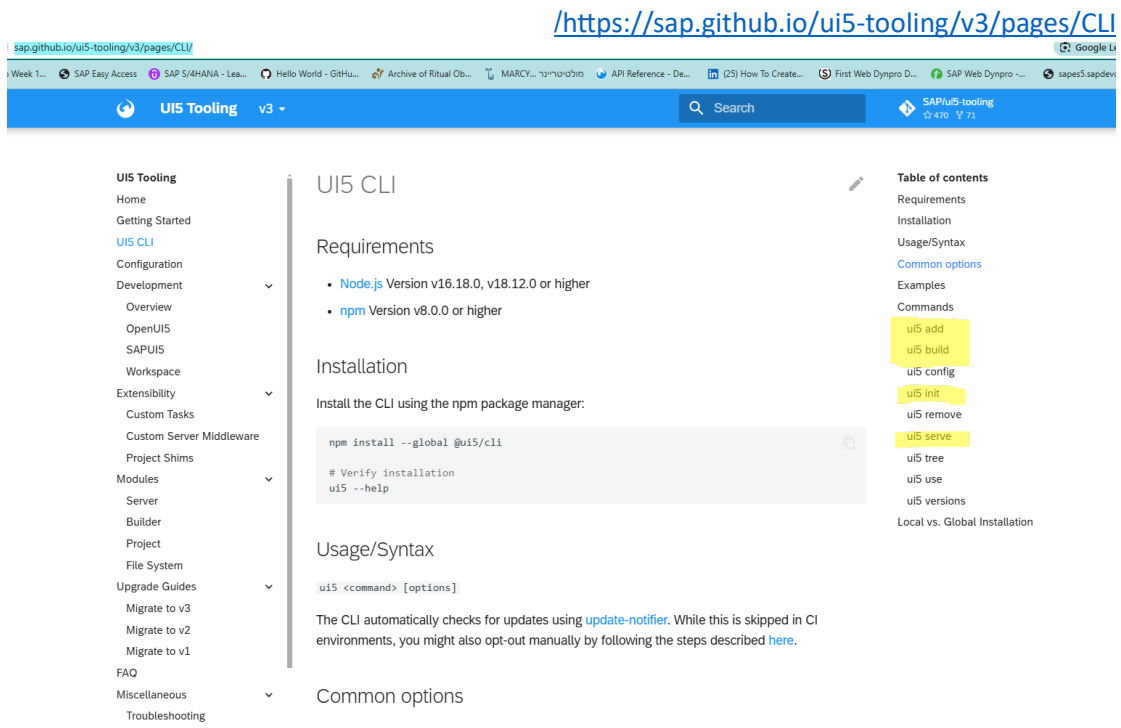


```

> ✓ TERMINAL
● user: projects $ cd project2
● user: project2 $ ui5 build
INFO: Using local @ui5/cli installation

info ProjectBuilder Preparing build for project project2
info ProjectBuilder Target directory: ./dist
info Project 1 of 1: > Building application project project2...
info project2 > Running task escapeNonAsciiCharacters...
info project2 > Running task replaceCopyright...
info project2 > Running task replaceVersion...
info project2 > Running task minify...
info project2 > Running task generateFlexChangesBundle...
info project2 > Running task generateComponentPreload...
info ProjectBuilder Build succeeded in 253 ms
info ProjectBuilder Executing cleanup tasks...
○ user: project2 $
```





So this is entire SAP web application bundled into this component.

Preload dot js.

When we run this Ue5 application in a browser.

Instead of loading these files independently, the browser is going to load component preload dot js.

This component preload dot js will have all the minified code.

See here.

This is all the minified code.

This is not the code which is there inside this app.

Dot controller.

Sorry.

Inside this particular controller.

So these are the original controllers.

App debug dot controller.

Dot JS is the original controller and this is the minified format you are not able to I mean generally

not many people will try to understand the minified format.

Only browser will load this minified format to improve the performance of the application.

So this is one more way of using Uf5 tooling.

So there are n number of commands which we can use for the UI for tuning.

If we come back here we can see Uf5 add is there Uf5 build.

So we have used Uf5 init Uf5 serve Uf5 build.

We can also try to use Ue5 ad as well.

So if we just click here, here are the examples.

If I want to use these libraries just copy this command here.

Come back to this project two and paste it.

And now press enter button.

So there is some issue it seems.

So it is suggesting us to use this command I believe.

So I just copied it.

Now just paste it here and try to use.

```
user: projects $ cd project2
● user: project2 $ ui5 use sapui5@latest
INFO: Using local @ui5/cli installation

Updated configuration written to ui5.yaml
This project is now using SAPUI5 version 1.131.1
○ user: project2 $
```