SAP Web IDE
Solution Overview

Provided by SAP Web IDE PM Team
September 2017
Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP’s strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP’s intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.
Agenda

• Introduction and overview
• Developing applications
• System Components
• Additional Information
Introduction and Overview
Key Trends and Impacts in Business and Technology

• Evans Data Corporation:” More than 95% of developers plan to use development tools in the Cloud on a SaaS basis next year…”¹

• IDC: “Cloud IDEs have matured for a broader range of software development scenarios.”²

¹ Source: Evans Data Corporation, Cloud Development Survey Volume I, 2016
SAP Web IDE is a powerful, extensible, web-based integrated development tool that simplifies end-to-end full-stack SAP application development.
Delivery Options

**CLOUD**

**Use case**
- SAP CP Web IDE
  - Develop UI applications

**Technology**
- Use OData Services:
  - SAP core, S/4HANA, SuccessFactors...
- Deploy to:
  - SAP CP (Neo) or ABAP

**Delivery**
- Zero installation, low administration
- Ubiquitous availability
- Biweekly updates
- Included with various SAP Cloud Platform packages

**HANA on-premise**

**Use case**
- SAP CP Web IDE Full-Stack
  - Develop and extend S/4HANA and SaaS applications

**Technology**
- Use OData Services:
  - SAP core, S/4HANA, IoT, SuccessFactors...
- Deploy to:
  - SAP CP (Neo, CF) or ABAP

**Delivery**
- Zero installation, low administration
- Ubiquitous availability
- Biweekly updates
- Included with various SAP Cloud Platform packages

**SAP Web IDE for SAP HANA**

**Use case**
- Develop native applications for SAP HANA

**Technology**
- Use OData Services:
  - Node.js, XSJS, Java
  - SAP HANA

**Delivery**
- Released with SAP HANA SP5s
- Included in SAP HANA license
- Available with SAP HANA, express edition
Key Use Cases

Build & extend SAP Fiori / UI applications
- Leverage out-of-the box code templates based on SAP best practices
- Extend existing SAP Fiori applications via predefined extension points
- Adapt the user interface of SAP Fiori elements and applications
- Use text editors and graphical tools
- Easily deploy to various SAP platforms

Develop and extend SaaS and S/4HANA
One end-to-end integrated development experience for all of the application’s modules
- Create an SAP HANA data model
- Develop Java-based business logic
- Consume the service with SAP Fiori UI

Develop mobile applications
- Create, test, build and deploy hybrid mobile apps
- Codeless, metadata-driven application development
- Leverage mobile services integration

SAP Leonardo (IoT) applications
- Jumpstart development with SAP Leonardo application builder with RAD capabilities and IoT-specific templates
- Use predefined SAPUI5 components targeted for IoT
Developing Full-Stack Business Applications

SAP Web IDE

User Interface

Business Logic

Data Models & Analytics

SAP Cloud Platform
Cloud Foundry
Code Simple

Key capabilities

- Zero installation and upgrade
- Rapid app development
- Full-stack application development, extension and deployment – UI, business logic and database modeling.
- Supports SAP technologies (Fiori, Leonardo IoT, HANA, …) and open-source (JS, OData, Java) technologies
- Integration with various data sources
- Integrated Git source control
- Extensible and modular architecture

Key Benefits

- Reduces cost, complexity and effort
- Increases developer productivity
- Improves team productivity
- Develop once, deploy to SAP platforms: SAP Cloud Platform, ABAP, SAP Mobile Platform and SAP Enterprise Portal
Additional Features

- Sample applications, templates and wizards for creating SAP UI5 applications, SAP Fiori applications, SAP HANA database modeling and Java applications
- SAP HANA database graphical editors
- Creating and testing applications with specific SAPUI5 version
- Source code editor with code completion, validation and formatting for JavaScript, SAPUI5 and Java
- Layout Editor (WYSIWYG) for SAPUI5
- User-friendly graphical extension mechanisms for Fiori applications
- Built-in application preview
- Mock data support for backend-decoupled development and testing
- Integrated deployment to SAP platforms (SAP Cloud Platform, ABAP repository, …)
- SAP Web IDE extension possibilities via Features
Out-of-the Box Features

**SAP Leonardo (IoT) Application Enablement**
Build and deploy new IoT applications and customize them by using predefined components and templates.

**SAP Build**
Comprehensive set of tools that enables organizations to design and prototype enterprise apps that make end users happy.

**SEAM & Hybrid Application Toolkit**
Create, package and deploy native apps using codeless tools or hybrid mobile SAP Fiori apps based on Apache Cordova and Kapsel.

**SAP BPM Workflow**
Rapidly build and extend workflows with easy-to-use graphical modeling tools and directly deploy them to SAP Cloud Platform workflow service.

**SAP API Business Hub**
Discover and consume SAP cloud services during design time. Test the app in a sandbox environment with mock data.

**And more…**
For a selected list of SAP Web IDE out-of-the-box features, visit SAP App Center or SAP Web IDE documentation.
Developing Applications
Welcome Screen

Perspectives:
- Home (Welcome)
- Development
- Database Explorer
- Learning Center
- Preferences

- Current version
- Notification every time an update is pushed.
- Users can read about new features

Quick and easy access to the most common creation actions

Notifications
- User Name
- Logout

Link to external resources and tools
Development Perspective Overview

Menu Bar

Global Toolbar

Workspace:
- User’s copy of the projects

Editor/Layout Editor:
- Code editor
- Code completion

Bottom Pane:
- Show/hide various panes, e.g. Console, Problem View etc.

Right Side Pane:
- Show/hide various panes, e.g. Search, Git, Gerrit …
Development Perspective Overview – Continued

Search:
- Advanced Repository Search
- Find in files (file name or content)
- Find references

Git:
- Manage your development objects in your Git repository

Gerrit:
- Manage your code reviews

Git History:
- Tag, Cherry-Pick, Revert etc.

Outline
SAP Web IDE Features

- SAP Web IDE is a modular, extensible plug-in framework. **Features** are constructed from one or more plug-ins, and each encapsulate different capabilities, such as development scenarios, templates, tools and more.

- You can enable features in Settings.

- Perspectives, views, menu items etc. are configured according to the enabled features.

- For example, you must enable “Tools for Java development” to be able to create a Java module.
Multi-Target Applications

- A multi-target application (MTA) is an application that is comprised of multiple software modules representing the data, business logic and UI.
- The modules can be created with different technologies.
- The modules share the same development lifecycle.
- An MTA can be deployed to different target platforms.
Development Process Overview

1. **Create**
   - Start with creating a new project using quick start, templates, sample applications and more, or import an existing project from Git, an archive or other repositories.

2. **Develop**
   - Continue development using optimized code editors and rapid development tools such as layout editor, graphical application extension tools and more.

3. **Build & Test**
   - Build your project and modules, and test your application with unit tests and mock data.

4. **Deploy**
   - Deploy your application to one or more of SAP platforms – SAP CP Cloud Foundry, SAP CP Neo, ABAP repository and more.
Create a New Project

To create a new project, you have the following options:

New Project from Template
Several templates for SAP Fiori and UI5, such as SAP Fiori master details and worklist app, as well as a template for creating a multi target application project.

Sample Application
Three complete SAP Fiori apps, available as reference, including mock-server support.

Quick Start
Creates a simple model, view and mock data. Launches the Layout Editor.

New Extension Project
Creates a new extension project to extend a Fiori application.

New from SAP Fiori Cloud
Launches SAP Web IDE and creates an extension project directly from the Fiori cloud edition.
Project Templates

- The template wizard guides the user through the definition of an application and then generates the code.
- Standard templates are provided:
  - Multi-Target Application
  - SAP Fiori Applications: SAP Fiori Master-Detail application, SAP Fiori CRUD Master-Detail application, SAP Fiori Worklist application and more
    - SAPUI5 Application
    - SAP Web IDE Feature
- Additional templates of any kind can be created, added and managed.
Multi-Target Application Project
Full-Stack Development

• Full-stack application is a multi-target application (MTA) project with multiple modules.
• Every Application deployed to Cloud Foundry from SAP Web IDE should be an MTA.
• To create an MTA, create a new project from template and choose the Multi-Target Application template
• A new project is created with an MTA descriptor file (mta.yaml)
• Different modules can be created in an MTA project – HTML5 module, Java module and HDB module.
Sample Applications
SAP Fiori and UI Development

- SAP Fiori sample apps are complete apps with all code.
- They can be used as a reference or for copying & pasting code.
- They provide exemplary code and comprehensive comments, representing best practices for Fiori applications.
- Mock-server support allows immediate look & try even without a backend system.
- Three sample applications are available:

<table>
<thead>
<tr>
<th>Shop</th>
<th>Approve Purchase Orders</th>
<th>Manage Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>browse products and add them to a shopping basket</td>
<td>review and approve/reject purchase orders</td>
<td>maintain the product master of the Shop</td>
</tr>
</tbody>
</table>

- **Shop**: Transactional application, Full screen pattern
- **Approve Purchase Orders**: Transactional application, Master-detail pattern
- **Manage Products**: Transactional application with edit, Master-detail pattern
Quick Start with Layout Editor

SAP Fiori and UI Development

- Project “QuickStartApplication” is generated
- A model is created with three simple entity types
- JSON files with mock data are generated
- A View is created
- Layout Editor is launched with the created view.
SAP Fiori Extension Project
SAP Fiori Development

When creating a new extension project, you need to identify the parent SAP Fiori/SAPUI5 application being extended. You have two options:

1. The parent app is from ABAP repository
2. The parent app is from SAP Cloud Platform

Additional options in the wizard:

- An option to import the original application.
- An option to open the extension project in the extensibility pane after project generation.
- An option to choose the runtime SAPUI5 version. (Only when the parent app is from an ABAP repository).

The tool creates the extension project in your workspace and generates files such as

- Component.js
- index.html
- And more…
Importing Projects

You have the following options to import a project into your workspace:

**Import from Git**
Use Git clone to clone an existing project into SAP Web IDE

**Import from Archive**
Import an application from the file system (file, Zip archive etc.)

**Import from SAPUI5 ABAP repository**
Import an application from an ABAP repository

**Import from SAP CP**
Import an application that is deployed to SAP Cloud Platform

**Import from Build**
Import a project prototyped in Build
Development
Key Capabilities

Development Perspective
• Content-based search for names and resources across all the user's files and workspaces
• Problem View presenting issues in your workspace
• Personalization of code editor theme and font size

Code Editors
• Robust editors with capabilities of code completion (SAPUI5, JavaScript, XML, property files), code validation, code templates, and many more
• Java code editor with code completion and validations

Visual Editors
Layout editor, visual extensibility tool, SAP HANA database graphical editor, SAP HANA database explorer and additional visual editors enabling rapid and simplified end-to-end application development

Collaboration
Collaborate on application development using Git
SAPUI5 Development - Code Editors
SAP Fiori and UI development

• Code editor for JS, XML, JSON and CSS
• Integration with static code quality tools (ESLint)
• JSON validator – syntax, neo-app and manifest
• Semantic checks for SAPUI5 XML
• Beautifiers for JS, XML, JSON and CSS
• API reference pane for SAPUI5 control documentation
• Outline pane for the JS main entities and the flow of the application
• Cross-file navigation for JS
• Compare editor and manifest (AppDescriptor) editor
• SAPUI5 code completion for XML, JS-AMD and non-AMD
  ▪ Editor completes words, code fragments or entire SAPUI5 objects
  ▪ Cross-file code completion
  ▪ Snippet-based XML and JS auto-completion
  ▪ Easily set a different SAPUI5 version
  ▪ Auto-hint – code completion while typing
Design your view graphically in the Layout Editor and display the content of the XML view

- Right-click an XML file and open it with the layout editor.
- Design the view in the graphical display (canvas).
- Drag and drop the SAPUI5 controls from the palette to the Canvas.
- Configure the properties and events of the controls.
- Bind the data fields to the OData service entity sets.
Layout Editor (WYSIWYG)
SAP Fiori and UI development

Canvas
- Content of the XML view
- Graphical display

Palette/Outline
- SAPUI5 controls
- drag & drop controls to the canvas
- Add/Remove controls from Outline
- Focus on control from Outline

Device Form Factor
- Smartphone/Tablet/Desktop

Properties, Events and Data Pane
- Shows properties and data of selected control
- Entity sets and properties of OData service
- OData binding

Panes Hide/Show

Undo/Redo
SAP Fiori Extensions - Visual Extensibility
SAP Fiori and UI development

• While running the app, you can select an element either in the Outline pane or directly in the app when in Extensibility mode.

• Appropriate extension options can be chosen.

• Application can get data online from backend or use mock data.

Language:
• If the application supports multiple languages (i18n properties), then the language text can be changed on the fly.

Show all application elements, or filter by extensible or extended elements or extension points.

Outline pane:
• shows the UI elements available in the application, as well as extension points and UI controller hooks
SAP Fiori Extension Templates
SAP Fiori and UI development

- The template for extending an SAP Fiori resp. SAPUI5 app guides the user through a step-by-step process

- All extension options of the SAP Fiori / SAPUI5 extensibility concept are provided in the template:
  - Create extension project
  - Extend controller
  - Hide a control
  - Extend a view or fragment
  - Replace a view
  - Replace a service
  - Customize text of i18n resources
  - Implement a UI controller hook

- To add a new extension, select the extension project and choose one of the available templates
Develop SAP HANA Database Module
Full-Stack Development – Database Modeling

- Create a new SAP HANA database module in an MTA project:
  - The module will contain all of your artifacts.
  - A folder for your module, with an inner “src” folder is created in the MTA project and the mta.yaml file is updated with the module default parameters.

- Designated editors for SAP HANA artifacts:
  - HDBCDS textual and graphical editors
  - Calculation view editor
  - Function editor
  - Procedure editor

- SAP HANA database explorer
Develop a Java Module
Full-Stack Development - Business Logic

- Create a new Java module in an MTA project:
  - Choose one of the available templates for a Java application such as: Spring Boot application or Web application with OData V4 support.
  - A folder for your module is created in the MTA project and skeleton files are generated. In addition, the mta.yaml file is updated with the module default parameters.
- Continue developing using the Java code editor, which provides code completion and validations.
Create an HTML5 Module
Full-Stack Development – UI

• Create a new HTML5 Module in an MTA project
• Choose one of the available templates for an HTML5 application
• A folder for your module is created in the MTA project and skeleton files are generated. In addition, the mta.yaml is updated with the module default parameters.
UI Application Build
Grunt Build

- Grunt is an open-source task runner based on the Node.js® runtime environment. Grunt allows you to automate tasks that front-end developers perform on a regular basis, such as minifying JavaScript and CSS files, unit testing, linting files to check for errors, compiling CSS preprocessor files such as LESS or SASS.
- SAP Web IDE includes a Grunt task runner.
- Configure by:
  - Specifying dependencies in the `package.json` file.
  - Defining tasks in the `Gruntfile.js` file.
- You can use your own Grunt plugin or other publicly available plugins.
UI Application Build
Grunt Build – Continued

• To run the Grunt build, right-click on your project’s root folder and select **Build** from the menu or context menu. The build does the following:
  ▪ Runs the tasks specified in the *Gruntfile.js* file.
  ▪ Creates a new folder in your project named *dist* containing the build artifacts.
Full-Stack Application Build
Prerequisites

• Prior to building an MTA project you must:
  1. Select a space for building and running your project in SAP Cloud Platform, Cloud Foundry environment.
  2. Install a builder in the selected space. A builder is the component required for building the different modules.
• You can optionally reinstall a builder if you need to update the builder version.
Full-Stack Application Build
Building During Development

- While developing your application, you need to build the database and Java modules to test your work.
- Dependencies may exist between modules.
- Build your modules according to the dependencies. In other words, if the Java module is dependent on the database module, build your database module before building your Java module.
- The HTML5 module doesn’t need to be built during development as it contains only static resources.
Keep the following in mind when generating a build of the SAP HANA database module:

- SAP Web IDE supports the build of every SAP HANA database artifact.
- Select the module or the files you want to build, then choose **Build** from the menu or context menu.
- You must build the entire module on the first build. In subsequent builds, you may build only the changed files.
- Building the module creates the relevant SAP HANA artifacts.
Full-Stack Application Build
Building During Development

Keep the following in mind when building the Java module:

• Select the Java module, then choose the Build from the menu or context menu.
• Building the module compiles the Java code.
• Optionally, you may choose to build and run tests.
Full-Stack Application Build
Building Prior to Deployment

• Once you’ve completed the development of your MTA application, you need to build it.

• Building the application before deployment must include all the modules, including the UI module.

• The build process create a multi-target archive that packages all the modules for deployment.

• To build your MTA application:
  ▪ Select your project and choose **Build** from the menu or context menu.
  ▪ All modules will be built and an *.mtar* file will be created that is ready to be deployed.
Testing in SAP Web IDE
Test Authoring for UI Applications

• SAP Web IDE provides both wizard-based and code-based assistance for developing unit tests and integration tests for SAP Fiori and SAPUI5 apps.
• Create OPA pages, OPA journeys, and QUnit tests in a project using a wizard.
• Use code completion to add snippets for OPA tests in an OPA journey.
• HTML view for all test results with options to run coverage and filter results.
Testing in SAP Web IDE

UI Test Execution

SAP Web IDE empowers developers to test and evaluate their app’s functionality and performance during development, including the following:

- Instant app preview, with various presets
- Mock business data
- Execution of QUnit and OPA tests
Testing in SAP Web IDE
Authoring and Running JUnit Tests

• SAP Web IDE supports JUnit 4.
• Write JUnit tests in your Java module’s `src/test` folder.
• To run your Java module’s unit tests, right-click the module’s root and choose **Build and Run Tests** from the context menu. This runs all tests under the `src/test` folder.
• View results in the **Test Results** view in the right pane.
• Optionally, you can display the file coverage or export test results.
Testing in SAP Web IDE
Running a Java Application

To run your Java application:

Right-click your Java module and in the context menu, choose **Run Java Application**.

Results:

- Your application is built.
- The application is deployed to CF and then it runs.
- The output, state, and URL of the application are available in the Run console.
Testing in SAP Web IDE
Running a UI Application

You can run the application to test functionality, design and performance in any of the following ways:

- **Run as**
  - **Web Application**: Previews the application in a browser.
  - **Unit Test**: Runs the application as a unit test.
  - **SAP Fiori Component on Sandbox**: Runs the application in the SAP Fiori launchpad environment.

- **Run Configurations**
  Defines how the application runs, including the SAPUI5 runtime version.

- **Run with Mock Data**
  - Automatically generates data.
  - Generates and edits persistent mock data, providing mock data in JSON files.

**NOTE**: When running an HTML5 module, you may be prompted to insert your credentials in order to create the required destinations in CF.
Testing in SAP Web IDE

Run Configurations

You can configure how to run an application in the **Run Configurations** dialog box as follows:

- Specify the path to the file that is being used to run the application.
- Run the application preview with or without frame.
- Select **Open the application with mock data**.
- Select the application URL parameters and URL hash fragment.
- Specify the SAPUI5 runtime version.
- Specify the application destinations.

<table>
<thead>
<tr>
<th>Application</th>
<th>File Path</th>
<th>Frame</th>
<th>Mock Data</th>
<th>Application URL Parameters</th>
<th>SAPUI5 Runtime version</th>
<th>Application Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Application</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Unit Test</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>SAP Fiori Component</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Testing in SAP Web IDE
Application Preview in the Browser

Frame:
Option to run with Frame in “Run configurations”

Choice of resolution:
• Desktop (large)
• Mobile devices, e.g. iPad (medium), iPhone (small), Android, etc.
• Define a custom size
• The application is responsive – will adapt its appearance to the screen size
• Multiple screen sizes are predefined

Orientation:
• The orientation of the screen can be toggled between landscape and portrait view

QR Code generation:
• A QR Code is generated for the URL by clicking the rectangle

Language:
• If the application supports multiple languages (i18n properties), then the language text can be changed on the fly

Refresh:
• The application can be reloaded without needing to rerun it from the development environment

Live Preview:
• refreshes the preview automatically on change of resource
Testing in SAP Web IDE
Run with Mock Data

You can run the application with mock data, for example when your back-end systems are not ready or available. You can configure this in your run configuration.

The following options are available:
- Let the mock service generate data on-the-fly (metadata.xml).
- Let the mock service generate data that is stored in the workspace as JSON files (JSON files).
- Provide data in a JSON file.
- Add custom mock requests.

**Edit Mock Data** allows you to create or edit a mock data JSON file.
Application Deployment

Multiple options are available for a developer to deploy a new application or update an existing application on the target system:

- SAP Cloud Platform, Cloud Foundry Environment
- SAP Cloud Platform, Neo
- SAPUI5 ABAP repository, such as SAP Gateway, SAP Fiori front-end server
- Platform mobile service (for hybrid applications)
- SAP Enterprise Portal

Additional features may implement deployment to other platforms
Application Deployment

Application Status

• Check if your application has been deployed to the platform or SAPUI5 ABAP Repository. If not, you may directly deploy from here.

Additional options are available for SAP Cloud Platform:

• Check if your application has been registered to SAP Fiori launchpad. If not, register it.
• Once registration is complete, a new tile is created in SAP Fiori launchpad, assigned to a group (optional), category, and content package.
Deploy to SAP Cloud Platform – CF
Full-Stack Development

• Building your project is required before deployment
• The build process generates an .mtar file that packages your modules (applications).

To deploy your application:

1. Specify the .mtar file you want to deploy under the mta_archives folder.
2. Select the Cloud Foundry API endpoint, organization, and space.
• Applications packaged in the .mtar file are deployed to the CF space defined for your project.
Deploy to SAPUI5 ABAP Repository

To deploy an application to the SAPUI5 ABAP repository:

1. Choose the system you want to deploy to.
2. Deploy your application as a new application or update an existing application.
3. If required according to your development status and the chosen package, select a transport request. The request ID will be automatically generated.
Deploy to SAP Cloud Platform - Neo

To deploy an application to SAP Cloud Platform – Neo:

1. Choose whether to deploy as a new application or as an update to an existing application.
2. Select the account to deploy to.
3. Specify whether you want to commit and push to the platform’s Git repository.
4. Set the version number.
5. You may define whether the application should be automatically activated on the platform.

Once the application is successfully deployed, you may:

- Check the application status: State, Versions, Active Version.
- Register the application to SAP Fiori launchpad.
System Components
SAP Web IDE Architectural Overview
Connecting Cloud Applications with On-Premise Systems

Accessing on-premise systems from the cloud can be achieved in two ways:

**Usage of SAP Cloud Connector as on-premise agent**

- on-premise systems not reachable from Internet, easy setup and configuration.
- additional on-premise component.

**Reverse proxy approach**

- existing network infrastructure can be used.
- configuration of DMZ and firewall needed; attacks from Internet possible; not all protocols can be supported.
Secure Integration with SAP Web IDE, Business Suite, and Other On-Premise Systems

- Establishes secure SSL tunnel between the SAP Cloud Platform and on-premise systems
- Connectivity created by on-premise agent through reverse-involve process
- Supports pre-configured “Destination API” and certificate inspection to safeguard against forgeries
- Complementary to SAP Gateway, SAP Cloud Platform Integration and 3rd party integration suites both on-premise and in the cloud

For more information on SAP Cloud Connector please refer to whitepaper [http://scn.sap.com/docs/DOC-60903](http://scn.sap.com/docs/DOC-60903)
Running HTML5 Apps on SAP Cloud Platform (Neo)

Developer / Designer / Functional Specialist

SAP Web IDE CP

Deployment Infrastructure

Push / Fetch

Git (App Projects)

HTML5 Apps Dispatcher

User Workspace

HTTPS

SAP Gateway

Via Cloud Connector

Trust RFC

Business Suite / ERP

Corporate Network / Intranet

Internet / Cloud

Application User

neo-app documentation
Running HTML5 Apps on SAP Cloud Platform (CF)

- Developer / Designer / Functional Specialist
- SAP Cloud Platform
  - SAP Web IDE CP
  - Database
  - Development Infrastructure
  - User Workspace
  - Cloud Foundry
  - Neo
- Development Space
  - Builder
  - App
  - App Router
- Cloud Controller
- Internet / Cloud
- Corporate Network / Intranet
- SAP Gateway
  - Via Cloud Connector
- Business Suite / ERP
- Application User
Git Source Control

What is Git?

- A widely used and popular distributed version control and source code management system.
- Records snapshots of all your files over time.
- Independent of network, user works offline (contrary to ABAP).
- No exclusive locking but conflict detection and powerful merge algorithms/tools.
- Powerful branching options allow developing several projects independently.

SAP Web IDE has a built-in Git client
SAP Web IDE and Git Integration

SAP Web IDE includes integration with the Git source control system, via its built-in Git client

- The SAP Web IDE Git client can work with SAP Cloud Platform Git and with any public Git server, such as GitHub, Bitbucket.
- SAP Web IDE provides a UI for executing Git commands and managing your source control and versioning.
  - Git pane for common commands.
  - Git History pane for additional commands.
  - Gerrit pane for viewing and submitting changes.
- Conflict resolution - graphical Git Rebase Interactive using integrated compare editor.
Git on SAP Cloud Platform used by SAP Web IDE

- Source code is stored in Git repositories.
- Every developer has a local clone of the central remote repository created by the **clone (1)** operation.
  - Clone is only executed initially to create the local Git repository.
  - If clone is executed again, it overrides the local Git repository.
- A specific state of the repository is **checked out (2)** into the developer’s workspace.
- Changes are collected in **commits** (≈ ABAP transport); that is, the developer **commits (3)** his or her changes to the local Git repository.
- Commits can be **pushed (4)** to and **fetched (5)** from the remote repository.
Additional Information
Useful Links

SAP Web IDE

• SAP Web IDE Home Page: http://developer.sap.com/webide
• Technical Question – Use SAP Community: https://www.sap.com/community/about/questions-and-answers.html
• Contact the SAP Web IDE team: sapwebide@sap.com
• Have an idea for a new feature or capability: https://ideas.sap.com/
• SAP Web IDE at SAP Store: https://www.sapstore.com/solutions/60009/Web-IDE

Additional information related to SAP Web IDE

• SAP Cloud Platform: https://cloudplatform.sap.com/index.html
• SAP Cloud Platform trial: https://account.hanatrial.ondemand.com/
• SAP Cloud Connector: https://help.hana.ondemand.com/helprameset.htm?e6c7616abb571014b75d96d596.html
• SAP Gateway: http://help.sap.com/nwgateway20

SAPUI5

• SAPUI5 Developer Center- Demo Kit (interactive doc on SAPUI5 / OpenUI5): https://sapui5.hana.ondemand.com/sdk/
• OpenUI5: http://sap.github.io/openui5/

SAP Community

• SAPUI5: https://www.sap.com/community/topic/ui5.html
• All Things SAP Fiori: https://wiki.scn.sap.com/wiki/display/Fiori/All+Things+SAP+Fiori
• SAP for Mobile: https://www.sap.com/solution/mobile-technology.html
Thank you.

Contact information:
Raz Korn, raz.korn@sap.com
Liat Borenshtein, liat.Borenshtein@sap.com
SAP Web IDE Product Manager