Single Page Application

COMP 3104

Week05

What is Single Page Application (SPA)

- A single-page application is an app that works inside a browser and does not require page reloading during use. For e.g. Gmail, Google Maps, Facebook or GitHub applications are some names.
- SPAs are all about serving an outstanding UX by trying to imitate a “natural” environment in the browser
  - no page reloads
  - no extra wait time.
- It is just one web page that you visit which then loads all other content using JavaScript – which they heavily depend on.
- SPA requests the mark-up and data independently and renders pages straight in the browser.
- SPA uses the advanced JavaScript frameworks like AngularJS, Ember.js, Meteor.js, Knockout.js .
- Single-page sites help keep the user in one, comfortable web space where content is presented to the user in a simple, easy and workable fashion.

Pros of the Single-Page Application

- SPA is fast, as most resources **(HTML+CSS+Scripts)** are only loaded once throughout the lifespan of application. Only data is transmitted back and forth.
- These makes development very simplified and streamlined as there is no need to write code to render pages on the server.
- SPAs are easy to debug with browsers, as you can monitor network operations, investigate page elements and data associated with it.
- It's easier to make a mobile application because the developer can reuse the same backend code for web application and native mobile application.
- SPA can cache any local storage effectively. An application sends only one request, store all data, then it can use this data and works even offline.

Cons of the Single-Page Application

- It is very tricky and not an easy task to make SEO optimization of a Single-Page Application. Its content is loaded by **AJAX (Asynchronous JavaScript and XML)** – a method of exchanging data and updating in the application without refreshing the page.
- It is slow to download because heavy client frameworks are required to be loaded to the client.
- It requires JavaScript to be present and enabled.
- Compared to the “traditional” application, SPA is less secure. Due to Cross-Site Scripting (XSS), it enables attackers to inject client-side scripts into web application by other users.
- Memory leak in JavaScript can even cause powerful system to slow down

Who Is a Front-End Developer?

- A front end web developer is the person who implements web designs through coding languages like HTML, CSS, and JavaScript.
- Front end developers are/were sometimes called "**client-side developers**" to distinguish them from back end developers who program what goes on behind the scenes like databases.
- If you head to any site, you can see the work of a front end developer everywhere–in the navigation, layouts including this article page, even the way that a site looks different from your phone (thanks to mobile-first or responsive design).
- Front end web developers use three primary coding languages to code the website and web app designs created by web designers:
  - **HTML**
  - **CSS**
  - **JavaScript**

DevOps Build Tool?

- Build tools are commonly known as programs that automate the process of building an executable application from source code.
- This building process includes activities like **compiling, linking, and packaging** the code into an executable form.
- Build automation involves scripting or automating a wide range of tasks that software developers perform in their daily activities.

The activities include:

- ❑ Downloading the dependencies.
- ❑ Compiling source code to the form of binary code.
- ❑ Packaging that binary code.
- ❑ Running the tests.
- ❑ Deploying them to the production systems.

Types of DevOps Build Tools

- SBT, is a build tool that is meant for Scala, Java, and many more programming languages.
- SBT provides a special interactive mode making Scala builds significantly faster using the same JVM as against the traditional batch operation where a number of build tasks are executed in a sequence.
- SBT is perfectly capable of compiling Scala code, packaging archive artifacts, executing tests, and also to support many other build operations.
- Following are some of the advantages of using SBT, let us now take a look at each one of them:

Advantages:

- SBT can be effortlessly used if it is a smaller and simpler project.
- Commonly identified being used with Scala open source projects.
- It provides an awesome integration if you are using IntelliJ IDEA IDE for your development.
- Most of the tasks that you will need (like compile, test, run, doc, publish-local, and the console) operations/tasks are usage ready with SBT.
- It is also pointed by few as a feature that dependencies can be open source repositories that are grabbed directly from GitHub.

Build Tools - Gradle

- Gradle is an open-source build automation tool focused on flexibility and performance. Gradle build scripts are written using a Groovy or Kotlin DSL.
- Gradle, an easier way to build your projects, has one of its most significant achievements to focus upon – the elimination of XML as part of the build script generation.
- Gradle uses a domain-specific language (acronym as DSL) that is based on Groovy, which is another programming language that can be run on the JVM.
- Gradle's DSL also lets you define both core parts of the build file and also specific steps called Tasks.
- The general Gradle builds files are appropriately named as 'build.gradle' and start with the very task of configuring the build. Having said all of this, you might be wondering did the Gradle team spent a whole lot of time trying to re-invent the wheel.
- Gradle proposed a fairly reasonable name called TASK for its build step as compared to ANT's obscure TARGET and Maven's confusing PHASE.

Gradle Advantages

- Adding dependencies is one of the most manageable tasks here, and also applying plugins.
- It is an entirely automated build process and also does provide an option to multi-module builds.
- Easier to use than ANT or Apache Maven and provides its support to finalizers, dry runs, and based on source changes automatically builds your project.
- Comes integrated with most of the popular IDEs like IntelliJ IDEA, Eclipse, etc via plugins.
- It can sense the changes in Gradle files and updates the Project structure accordingly.
- It is unbelievable that the learning curve on this tool is pretty flat.

Build Tools – Apache Maven

- Maven is a build automation tool used primarily for Java projects.
- Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages.
- The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.
- Apart from being just a build tool, Apache Maven is also a software project management and comprehension tool.
- Based on the concept note of **Project Object Model (POM)**, it can manage a project's build, reporting, and documentation processes from a central piece of Information.
- Apache Maven is two tools made into one – the Dependency Manager and also a powerful build tool.
- Like Apache ANT, it is an XML based build file.

Apache Maven Advantages

- Possibility and also ease in configuring the whole project by just glancing through the one important file of the whole project, the pom.xml.
- Apache Maven reduces the burden of keeping your dependencies up to date, as the dependency management module of Apache Maven kicks in to perform all such activities.
- An added benefit is the ease with which you can comfortably build a Cucumber project or a TestNG project.
- Once the configuration is done, the developers have only one task to do – Development with peace of mind!
- Project management skills can be seen very clearly when Apache Maven is let to handle a relatively bigger project for all its needs – build automation, dependency management and etc.
- It provides support to any platform for the actual build process.
- It provides wonderful support for all the Unit testing needs and logging requirements.
- Profile-based support to run the pom configurations based on the profile that it gets executed with.

What is Jira ?

- **JIRA** is a tool developed by Australian Company Atlassian.
- This software is used for **bug tracking, issue tracking,** and **project management**.
- The name "JIRA" is actually inherited from the Japanese word "Gojira" which means "Godzilla".
- The basic use of this tool is to track issue and bugs related to your software and Mobile apps.
- It is also used for project management.
- The JIRA dashboard consists of many useful functions and features which make handling of issues easy.

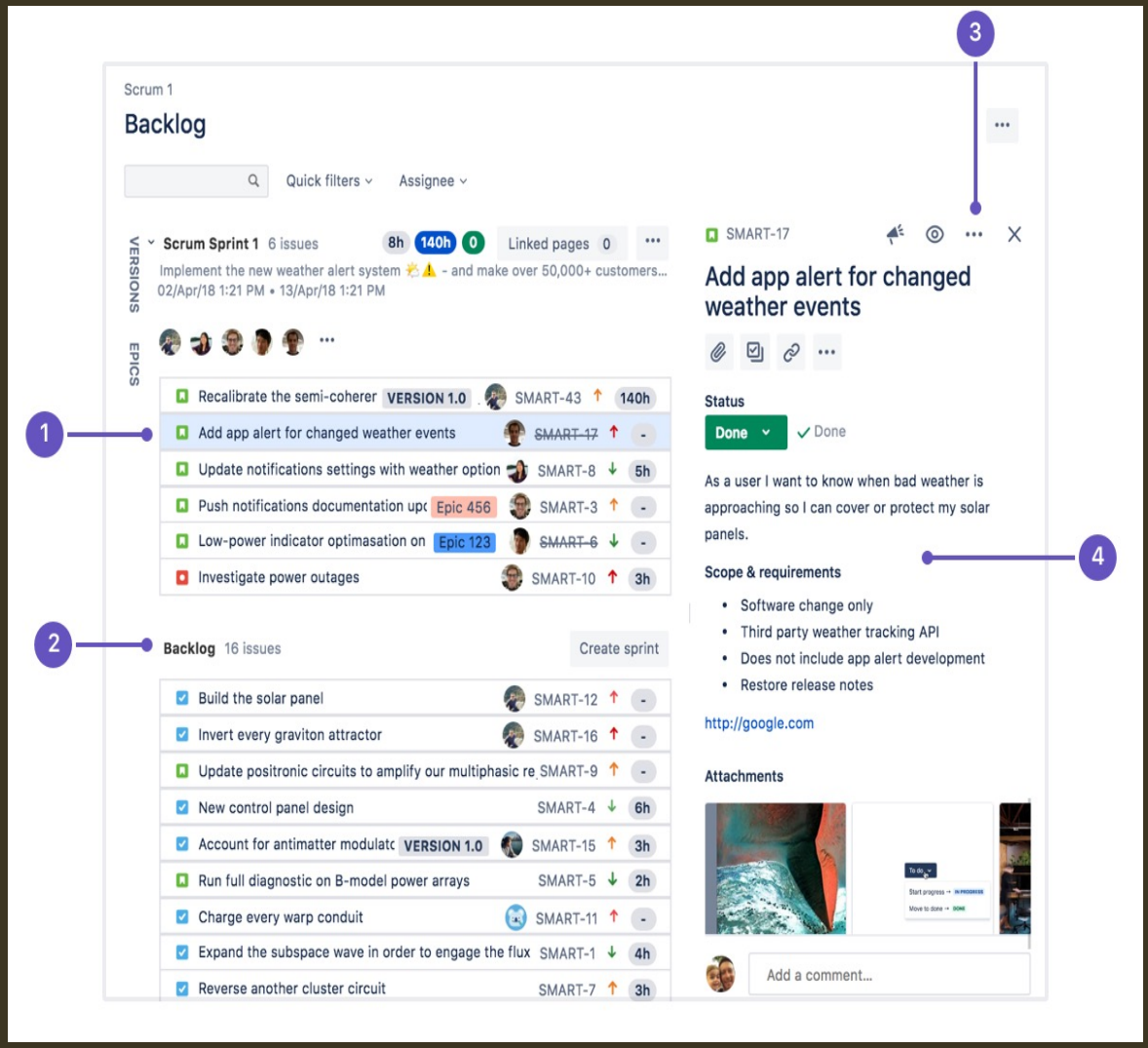
Click here to get more info on Jira

What is JIRA Issue?

Issues are the building blocks of any Jira project. An issue could represent a story, a bug, a task, or another issue type in your project.
JIRA issue would track bug or issue that underlies the project.

Under Issues, you will find other useful features like

- ❑ Issue Types
- ❑ Workflow's
- ❑ Screens
- ❑ Fields
- ❑ Issue Attributes



1. **Selected issue:** Select an issue to view its details.
2. **Backlog:** Estimate issues and plan your sprints.
3. **Issue actions:** Move an issue, clone it, log time spent working on it, and perform other actions.
4. **Issue details:** See the assignee and description, make comments, and add content to the issue.

Thank You