**Henry Chen**

**Summary**

• Over 5 years of experience as a Java full stack web Developer, creating **Single Page Web** applications using **Angular 2/4/5/6/7**, **Node.js** and **React.js** with backend using Spring MVC, Spring Boot, Hibernate and JPA in various domains.

• Proficient in developing Single Page Applications (SPA) using **Angular 2/4/5/6/7** and **React.js**.

• Develop SPA by using Angular related technologies, such as **TypeScript**, Angular **Components, Directives**, **Pipes, Services**, **HTTP Client**, **Observables**/**Promise**, **Reactive/Template** driven forms, **Two-way data binding**, **Life Cycle Methods**, **Routing** and **Routing Guards** within multiple components.

• Develop SPA by using React related technologies, such as **JavaScript ES6**, **Redux**, **CSS Module**, **Dynamic Styling**, **Axios**, **React Components**, **Props**, **States**, **Two-way data binding**, **Promise**, **Conditional Expression**, **Life Cycle Methods**, **High Order Component**, **Props Validation** and **Routing** within multiple components.

• Extensive experience to use front-end technologies like **HTML5**, **CSS3**, **JavaScript**, jQuery, **AJAX** for building dynamic, user-friendly, high performance web applications.

• Apply **responsive web design** to fit all users and devices from desktop to tablet and mobile phone by using **Bootstrap** and **Media queries.**

• Extensive experience in Backend Server-Side Programming using **J2EE technologies**, like **Java Core**, **Spring Core**, **Spring MVC**, **Spring Boot**, **Spring AOP**, **XML**, **Hibernate**, **JPA**, **JDBC** and Some other technologies Node.js, Express.js, LAMP, Laravel, Django.

• Extensive experience in developing **RESTful API** using J2EE Technologies and testing **APIs** using **Postman**.

• Expertise in database system like **SQL Server**, **Oracle**, MySQL and NoSQL database **MongoDB.**

• Expertise in creating Dynamic SQL and perform CRUD and Other SQL operations using Relational Database like **SQL Server**.

• Expertise in Unit Testing by using **Karma**, **Mocha**, **Jasmine** in frontend, and **JUnit** in backend.

• Expertise in debugging and troubleshooting existing code using **JIRA**, Developer tools and **GIT** as Version Control System.

• Involved in all stages of Software Development Life Cycle (**SDLC**) as well as working in **AGILE** Methodologies based development environment, participating in spring/iterations and **SCRUM** meetings.

• Basic idea in cloud computing and **Amazon Web Services** (AWS), **Microsoft** **Azure**, and **Google Cloud Platform**.

**Skills**

|  |  |
| --- | --- |
| ***Languages*** | Java, Python, JavaScript, TypeScript, PHP, C#, C/C++, R, shell scripts, SQL |
| ***Databases*** | SQL Server, Oracle, MySQL, MongoDB, SQLite, Redis |
| ***Web Technologies*** | HTML, CSS, Bootstrap, JavaScript ES5/ES6, jQuery, AJAX, Angular 2/4/5/6/7, React.js, Redux, SASS, Gulp, Webpack, Socket.io |
| ***Server-Side Technologies/ Frameworks*** | J2EE, Servlet, JSP, Spring Core, Spring AOP, Spring MVC, Spring Boot, Hibernate, JDBC, Maven, XML.  Node.js, Express.js, LAMP, Laravel, Django |
| ***Data Science*** | NLTK, Pandas, Numpy, Scikit-learn, Seaborn |
| ***Mobile Development*** | Android SDK, Android Studio, Gradle |
| ***Tools and IDEs*** | Eclipse, Visual Studio Code, Sublime Text 2/3, WebStorm, PyCharm, RStudio, PhpStorm.  Azure Data Studio/SQL Server Management Studio, Oracle SQL Developer, MySQL Workbench |
| ***Version Control*** | GIT, GitHub |
| ***Debugging/Testing Tools*** | Jasmine, Mocha, Karma, JUnit, Postman |
| ***Operating System*** | MAC OS X, Ubuntu/Linux Mint, Windows |

**Experience**

Project 1: Joole

**Responsibilities**:

• Contribute into a team in a **single page web application** (SPA) based on the **RESTful API** using **Angular 6/7**, **Spring MVC**, **Hibernate** and **SQL Server**.

• Used **React Form** to validate the user’s input in Login and Register Components.

• Used **Angular Routing** to navigate between different components.

• Utilized **Directives** in the application, such as **structure directives** \***NgFor** to dynamically exhibit multiple data entries, **attribute directives** like **NgClass** to add and remove CSS classes on an HTML element, **customer directives** to make reusable components.

• Write **Customized Pipes** based on business Logics by implementing PipeTransform interface.

• Used **Http Client Module** to send http request including GET, POST, PUT and DELETE to call Corresponding **Backend RESTful APIs** with Annotations in Controller Packages.

• Used **@input**, **@output**, **@Viewchild and local reference** for Component Communication.

• Used **Snapshots** to call the initial data and **Observables or Promise** to handle asynchronous data.

• Configured **Apache Tomcat 7.0/8.0/8.5** in Eclipse to start the server.

• Used **Docker** to start the SQL Server Database in the Mac, and **Azure Data Studio** for database graphic interface usage.

• Used **Maven** to handle and update dependencies, run **maven build**, **maven clean install** commands to start and refresh the backend applications.

• Configured **application.properties** file in resource packages for Database Connection and Hibernate Usage.

• Used **SpringMVC architecture** to build the backend project structure: **Domain**, **DAO**, **Service**, **Controller**, Exception.

• Used Hibernate **POJO** class for Object Relational Mappings (**ORM**) matched to database table records. JDBC is another option to interact with the database.

• Used **@CrossOrigin Annotation** in Controller Layer to fix front-end and back-end different ports data communications problems.

• Used **Karma** and **Mocha** to do front-end Angular Unit test.

• Used **JUnit** to do back-end API Unit test.

• Used **Slack** and **Email** to Communicate with team leader and members.

• Used **Git** and **GitHub** for Version Control.

Environment: **JavaScript, Typescript, HTML, CSS, Bootstrap, Angular 6/7, Git, Java Core, Spring MVC, Hibernate 5, Windows, MAC OS X.**

Project 2: Korea

**Responsibilities**:

• Developed a **single page web application** (SPA) based on the **RESTful API** using **React and Redux**, **Spring Boot**, **Hibernate** and **SQL Server**.

• Used **Redux** (**Action, Middleware, Reducer, Store**) for managing Application Global State.

• **Dispatch multiple actions** in Action layer and tackle actions in Reducer layer based on **action types**.

• Used **Axios** to send http requests to call backend RESTful APIs.

• Used React **Props** for Parent and Child Component Communication.

• Defined **a Global State** in each react components and **setState()** to update the component data.

• Used **React Life Cycle Hooks Methods**, such as ComponentWillMount(), ComponentDidMount(), ComponentDidUpdate(), render() in some components.

• Used **@trendmicro/react-sidenav** for Navigation bar.

• Used **<Route>, <NavLink>, <Link> tags** in react-router and react-router-dom for **Routing**.

• Broadly **Applied Spring Boot architecture** to build the backend project structure: **Domain**, **DAO**, **Service**, **Controller**.

• Implemented **POJO classes**, **JpaRepository interface**, **business logics**, **back-end API functions for front-end http requests** under Domain, Dao, Service, Controller layer respectively.

• Wrote **Customized Exception and utilities** in each created packages.

• Run the back-end Application with **a class have @SpringBootApplication Annotation** or using command **spring-boot:run**.

Environment: **JavaScript ES6, HTML, CSS, Bootstrap, React, Redux, Git, Java Core, Spring Boot, JPA, Windows, MAC OS X.**

Project 3: Pure Java Project Banking System

**Responsibilities**:

• Designed a J2EE Web application with Spring Core, Spring MVC architecture, JSP, Servlets, jQuery, Hibernate and Oracle.

• Coding both front-end and back-end inside Eclipse under the created Dynamic Web Project structure by the project manager.

• Configured Build Path by adding required external JARs for specific development.

• Put All front-end JSP pages, JavaScript, CSS, images under src/WebRoot directory

• Utilized form submission or jQuery Ajax calls to call the required APIs inside Controller layer.

• Developed back-end MVC architecture with Domain, Dao, Service, Controller layers.

• Write POJO classes, hibernate CRUD operations, business logics, back-end API functions for front-end http requests under Domain, Dao, Service, Controller layer respectively.

• Applied Servlet Life Cycle Methods init(), service(). doGet(), doPost() for Controller layers. Realized page redirection by using request.getRequestDispatcher().forward() with HttpServletRequest Class.

• Configured hibernate.cfg.xml for database connection settings with several properties inside <session-factory> tag

• Realizing Object relational Mappings (ORM) patterns by Mapping each POJO class to a corresponding Hibernate Mapping file(hbm.xml), Or use Annotations inside POJO classes.

• Extensively applied other Hibernate and database management technologies like Sessions, transactions, criteria queries, Hibernate Query Language (HQL), Native SQL, Cache, Batch Processing, Interceptors, to interact with the database.

• Initialized Spring beans and Called the bean xml files path by creating a new instance of ClassPathXmlApplicationContext class. Get the required bean with getbean() method under ApplicationContext class.

• Developed some Spring customized Dependency Injections based on business logics.

• Applied PL/SQL for complex DB operations, wrote SQL stored procedures for repeated operations.

• Applied JUnit tool for writing Unit /Integration test cases

• Used JIRA to track and report the issues in the application.

• Applied Git for Version Control and code management.

Environment: **HTML, CSS, Bootstrap, JSP, Servlets, Git, Java Core, Spring Core, Spring MVC, Tomcat 7.0, Hibernate 4, Windows 7, Oracle 11g.**

Project 4 : Node.js Project Pets System

• Designed a Single Web Application using **React**, **Redux**, **Bootstrap**, **CSS**, **Node.js**, **Express.js**, **MongoDB** and **mongoose**, **Google API**.

• Utilize **passport** to Call Google APIs for authentication if the users selected to login with Google Accounts.

• Used **Redux** (**Action, Middleware, Reducer, Store**) for managing Application **Global State**.

• Designed reusable and feasible component **containers** for inserting different components pages.

• Used React **Props** for Parent and Child Component Communication.

• Defined **a Global State** in each react components and **setState()** to update the component data.

• Implementing **Express Routing** by called get(), post(), put() and delete().

• Handle libraries with **NPM** tools, initialized package.json file with npm Init, run the application with **NPM start** command by configuring the start scripts setting.

• Configured **MongoDB Connections** and implemented **CRUD operations** with mongoose.

• Utilized JavaScript Asynchronous Mechanism with **Promise** or **Asyn/Await**, Handle Exceptions with multiple **try catch blocks** for Asyn/Await usage.

• Applied **JavaScript ES6 features**, for example, use **let** and **const** for variable definition, import packages with required() method.

Environment: **HTML, CSS, Bootstrap, React, Redux, Git, Node.js, Express.js, MongoDB, Visual Studio Code, Ubuntu 16.04.**

Project 5: Android Project Communication social media

• Developed a Native Android APP by using **Android SDK, LAMP**.

• Develop UI layout in xml files with **LinearLayout**, **RelativeLayout**.

• Applied UI Controls inside UI layouts for designing different pages, like **TextView, EditText,Button, ImageView, CheckBox, RadioGroup/RadioButton, AutoCompleteTextView, Spinner**.

• Implemented **Activity Life Cycle** Methods **OnCreate(), OnStart(), OnResume(), OnPause(), onStop(), OnDestroy() and onRestart()** in UI Components.

• Binding UI layouts and Activity class with **setContentView()** method inside OnCreate().

• Selected usage of **Activity**, **Services**, **Broadcast Receivers** and **Content Provider** based on business logics.

• Wrote customized Animation, Color Style, Images, Menu, Values inside **resource** package.

• Utilize **Handler** and **switch case blocks** to send and process different thread’s Message from MessageQueue.

• Use putExtra() and getExtra() methods inside the **Intent** class for **Component communication**.

• Developed **server-side APIs** using PHP inside Sublime Text. Called mysql\_connect() method for PHP MySQL connection.

• **Retrieve JSON data** by **calling server-side APIs** with **URLConnection** class in UI Components.

• Configured **AndroidManifest.xml** by adding multiple activities, uses-permission. Change the launching activity by move the <intent-filter> tag inside into different activities.

• Designed **customer Adapter** by extending BasedAdapter. Use setAdapter() method to bind the listView with the customer Adapter.

Environment: **Java Core, Android, MySQL, PHP 5.6, Apache server, Android Studio OR Eclipse (Android SDK, ADT plugin installed), Sublime Text 2 , Navicat, Windows 7, Ubuntu/Linux Mint.**