

BBC London Assessment Centre - Technical Questions

Name: Hongyi Duan

Technical Questions:

1. We're looking for people with a real passion for collaboratively creating great software. Please give an example of a software component you have designed and written from Concept to deployment, outlining the steps you took. (1000 character limit)

I have designed lots of software applications included native iOS/Android apps, web applications, Java Program Projects with WindowBuilder GUI and some small tools in the past 2 years in my spare time. The recent project impressed me the most is a web application Coupons and Orders Management Server. It is not a private project, the idea is started by a Chinese start-up Chinese start-up which proves online study videos at the end of 2016. I made about 2/3 contribution to this project. The Github link for this project: <https://github.com/LennyDuan/couponsys>.

In summary, our responsibilities (another developer and me) were to build a server which provides coupons and manages orders, and present the data to the main web-server using a Restful API. I developed in a test-driven way using node.js and MongoDB and had a great test coverage of Models, DB Proxy and API. My server also provided a responsive front end for verified administrators to implement basic create, retrieve, update and delete (CRUD) operations.

The developments steps:

1. Analyse the Requirements: Before this web server, we already have a main website which provides online video for users. The founder wanted some new features like provide the flexible coupon codes for sales and record the order details for managers but won't influence the user-side video website. So we decide to make a separate web server which provides APIs for the main website.
2. APIs design: According to the requirements, the most important part is API. So we use RAML to design and build our API. It is a really powerful tool for design multi complicated API before development. We pre-define the API paths, short descriptions, request params, request bodies and return values. We made little changes when we implemented these APIs but it helps a lot and guides the direction of development at the beginning.
3. Test-driven development: We use TDD in the project and I really think it is a good way to build high-quality applications can meet the requirements. And, it will save lots of time if there are something changed. We create hundreds of test cases for DB Model, Proxy, APIs and etc before implements to make sure app works well as what we want.
4. Development process: TDD Test -> Application Structure Design -> Database Model -> DB Operation Proxy -> API -> Controller -> UI Design -> Deployment -> Usages Test. Also, during the development, we have written some documents and refactored code as well.

2. Using the example that you provided above, tell us about a significant decision you made to solve a technical challenge. Give details of technologies that you chose and why you chose them. (1000 character limit)

I have never had a real problem or blockers. Google always has an answer. So I just list some technologies we choose.

Before development:

- **RAML**: RESTful API Modeling Language makes it easy to manage the whole API lifecycle and I like its concise - only write what we need to define, reusable and human-friendly.
- **GIT**: A version control system for tracking changes in computer files and coordinating work on those files among multiple people. We have 2 developers and won't code in the same network. So we choose GITHUB as our host server and share our code as a public project.

Server-side development:

- **Node.js + MongoDB + Express**: A free and open-source JavaScript software stack for building dynamic web application. Because all components support programs written in JavaScript, it is very suitable for lightweight and rapid development. And MongoDB is a NoSQL database so it is very flexible when we wanna change out DB models.
- **Gulp + Promise**: Use Gulp, a toolkit for automating tasks in development workflow, to test our implementation. Use Promise to solve the call-back issues for asynchronous computations during development.

Client-side development:

- **Bootstrap + AJAX**: When I build a website, I will always do the responsive UI design. Bootstrap is the great framework for responsive web application. As a user, I really don't like to refresh the page for small changes. So I will always use AJAX in RESTful HTTP requests.
- **CDN**: I use CDN to improve the response speed for external JS and CSS like Bootstrap, JQuery and etc. Actually, we have an issue for this decision. Because we develop the website in the UK, but all our Chinese teammates can't access the pages. Then I notice that 'Network Great Wall' in China will block sources from some site host.

3. Using the example that you provided above, tell us about how you ensured your software was fit for purpose and of high quality. What did you learn and what would you do differently next time to do a better job? (1000 character limit)

TDD is the certificate to ensure my software was fit for purpose and of high quality. Before development, I analyse the requirement first and then design APIs via RAML. Before implementing the features, I create lots of tests and have a good test coverage for different scenarios.

In this project, I learn a lot about Node.js, MongoDB, Gulp and TDD, Call-back and Promise and etc. But the most important is the suitable workflow to develop a website and the hierarchical design for a web application.

Next time, I will still use TDD for a software development. For a web application, I will learn new technologies and might use Angular.js (MVVC), React.js (View) or Vue.js as a client-side framework for a quick development. Ruby + Postgres could be a good choice as well. But the technologies are just tools, the most important are the requirements and choose the suitable tool for the different purpose.

4. Please tell us about the ideal position you are looking for and also confirm your key technical skills? Please use this opportunity to tell us about a position you most prefer so we can ensure you are matched into the right position should you be successful (500 character limit)

The position is the software developer. Actually, I can't make a detail decision right now. Could be front-end development, back-end development, Mobile Apps development, Gaming development or even algorithm and deal with big data.

Actually, my hobbies prove to be kind of too extensive rather than specialized like a full-stack developer. I have lots experience of native Mobile App development, web application, and traditional software projects.

My key technical skills: OOP, Java, Node.js, Javascript, front-end and iOS.

Other key skills: Clean Code, Test, GIT, Vim, Markdown