

Angular Developer Questions

Here's a bit of my script for interviewing people. Feel free to steal or provide suggestions to make it better... Admittedly, I'm a bit more in-depth on the front end side, specifically Angular 2+ since that is my specialty.

Angular 2+

- In general, describe a SPA architecture and how it differs from traditional server side MVC, and any advantages/disadvantages
- In Angular 2+ we have different file types- modules, components, templates, services, and classes (or interfaces). Can you describe the role that each file type plays and how everything comes together?
- Angular 2+ organizes your application as a tree of components. If I have a parent component with two children, and want to pass data down to the children so that they don't have to repeat a network call on initialization- what is a strategy to do that? How about passing data back up to the parent?
- Can you describe the lifecycle methods of an angular component and when each is called?
- Can you describe angular component change detection?
- In a typical angular component pretty straightforward to wire listeners to click events for the contents of the template. However, how would you listen to a click event in the component on the tag that actually represents the component (this tag occurs in the dom right before the first line of the template, and exists in the parent component's template)?
- What is a ViewChild decorator and can you explain some of the benefits?
- Can you describe the concept of a Promise, describe an Observable and explain the difference?
- Can you describe how you've used any rxjs libraries with Observables?
- Let's say you have a UI component that relies on two separate network requests. You'd like to send those requests off in parallel but wait to update the UI until both have resolved. What is a strategy for doing that?
- How would you handle an error return code from the backend? Both a business error where you need to inform the user of the business error failure and a fatal 'backend service is down' error?
- Can you describe how to apply custom validation to a form element in angular and how you'd supply an error message to the user directly underneath the input tag?
- Are you familiar with reactive forms and what advantages they provide over template driven forms?
- Can you describe some of the tools that are under the hood of @angular/cli and what it does for you?
- Can you describe the concept of a jasmine spy, when and how you would use it to write a unit test?
- Can you describe a strategy for testing an asynchronous operation in a jasmine unit test?

HTML/SASS/LESS/CSS/Some general javascript

- Are you familiar with the term semantic markup? How can I avoid 'div-itus', and why is using a div for everything bad?
- Let's say I have two div's I'd like to display side by side in a two column layout instead of one on top of another. What kind of strategy can I use to achieve something like this without using a predefined grid package?
- Can you describe a CSS flexbox and what kind of problems it helps solve? Are you aware of floats and some of the challenges with using them?
- Have you used a css preprocessor like less or sass? Can you describe some of the advantages?
 - o Variables, functions or 'mix-ins', nesting structure, extension of base, loops
- How can I create an animation using CSS only and what is the advantage of using CSS for animations rather than javascript?
- Where in html would you place javascript tags to load the js, and what is the reasoning behind it?
- Can you describe some good programming practices in js and things to avoid?
- What are some tasks that should be done in getting a (SPA) javascript application ready to deploy to production?
- Can you describe some of the new features that were introduced in ES6?
- Can you describe some of the Javascript API changes that were introduced in HTML5?
- How do you enforce a consistent coding style of Javascript with a team of developers?
- Let's say there is a problem with your front end application. Can you walk through the inspector and talk about each tab and how you can use it to find out what is going on with your app?

Full Stack

- Can you describe the principles of RESTful service design?
- Let's say I have a backend database table called 'Car' which stores cars. What RESTful endpoints would I expose to do normal CRUD operations on the table? Describe how these would work over HTTP including url's and http verbs
- How would you design a service to return a business error for the car table, for example an error message on create when the number of wheels on a Car exceeds 4. Ideally, we would like to tell the user that the number of wheels should not be more than 4.
- Let's say this table has a million records. How would you design a pagination system to avoid a performance problem?
- How would you implement a filtering system so that the user can retrieve only 'red' cars that are of type 'suv'?
- How would you design the authentication/authorization for your application? How would this exchange work from front end/back end?

Node.js

- What does node.js do well, what makes it unique compared to other server side languages like Java or ASP.NET and why would you want to use it?
- What is callback hell and what are some techniques you can use to mitigate it?
- What is the preferred method of resolving unhandled exceptions in Node.js?
- What is typically the first argument passed to a Node.js callback handler, and why?
- What is the anatomy of package.json and can you explain how it works?
- What is your strategy for keeping a large node.js project organized from a directory and naming convention point of view? What is a barrel (index.js) and how does it work?
- Can you describe a stream (or EventEmitter) in Node and some of the lifecycle events?
- How would you perform file read/writes in node?

Microservices

- What is the principle of microservices and what are some advantages/disadvantages?
- What is an API Gateway?
- Describe some of the thought process that went into deciding what different parts of an application were broken up into different microservices. Was anything shared? What part was shared and how was that designed? How was the data persisted? Was any data shared across microservices?
- Describe a typical CI/CD pipeline, what would go into it, and why it is essential to productive microservice development
- Describe the concept of an app container like Docker, and why it pairs well with a microservice style architecture
- Describe the concept of a container orchestrator like Kubernetes, and what role it can play with microservices
- Describe the concept of serverless architecture and the advantages/role they can play with microservices