

Angular Environments

Add New Environment



Angular Environments

- In previous videos, we saw the use of environment files

- `src/environments`

Default environment

- `environment.ts`

- `environment.prod.ts`

Prod environment

`environment.ts`

```
export const environment = {  
  production: false  
  
  // add your custom environment configs  
  // ...  
};
```

`environment.prod.ts`

```
export const environment = {  
  production: true  
  
  // add your custom environment configs  
  // ...  
};
```

Run with environment configuration

- Run with the default, just give command

```
$ npm start
```

Use environment.ts

- Run with production configuration

```
$ npm start -- --configuration=production
```

Use environment.prod.ts

How does it work behind the scenes?

How are the filenames mapped to an environment?

How can I add a new environment?

Behind the scenes

- Environment configurations are defined in: `angular.json`
- Angular CLI tool added entries for "production" configuration in the file initially

The diagram shows a portion of the `angular.json` file with several annotations:

- A green callout points to the `"production"` key under `"configurations"`, labeled *Configuration name*.
- A red callout points to the `"fileReplacements"` array, which contains a single object. This object has `"replace": "src/environments/environment.ts"` and `"with": "src/environments/environment.prod.ts"` properties.
- A purple callout points to the `"with": "src/environments/environment.prod.ts"` line, containing the text: *During processing, for production configuration, will load the file: environment.prod.ts*.

```
angular.json
...
  "configurations": {
    "production": {
      "fileReplacements": [
        {
          "replace": "src/environments/environment.ts",
          "with": "src/environments/environment.prod.ts"
        }
      ]
    }
  }
...
```

Behind the scenes

- In the "serve" section of angular.json, define:

angular.json

Configuration name

```
...  
"serve": {  
  "configurations": {  
    "production": {  
      "browserTarget": "angular-ecommerce:build:production"  
    }  
  },  
  ...  
}
```

Referenced from command-line when we start the app:

```
npm start -- --configuration=production
```

```
...  
"configurations": {  
  "production": {  
    "fileReplacements": [  
      {  
        "replace": "src/environments/environment.ts",  
        "with": "src/environments/environment.prod.ts"  
      }  
    ]  
  }  
}
```

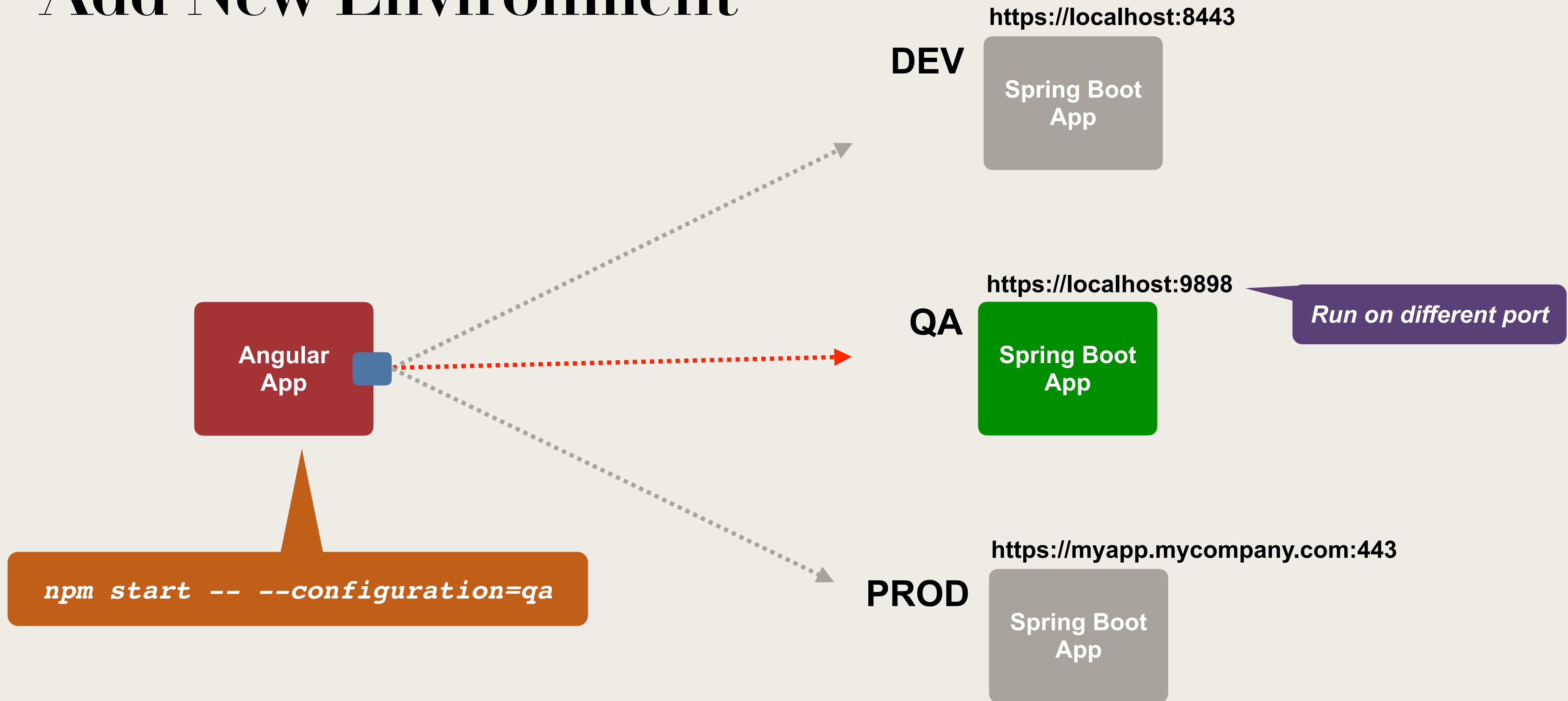
References the configuration defined on previous slide

~~How does it work behind the scenes?~~

~~How are the filenames mapped to an environment?~~

How can I add a new environment?

Add New Environment



Development Process

Step-By-Step

1. Create environment file: environment.qa.ts
2. Define configuration in angular.json for "build" section
3. Define configuration in angular.json for "serve" section
4. Run with environment configuration

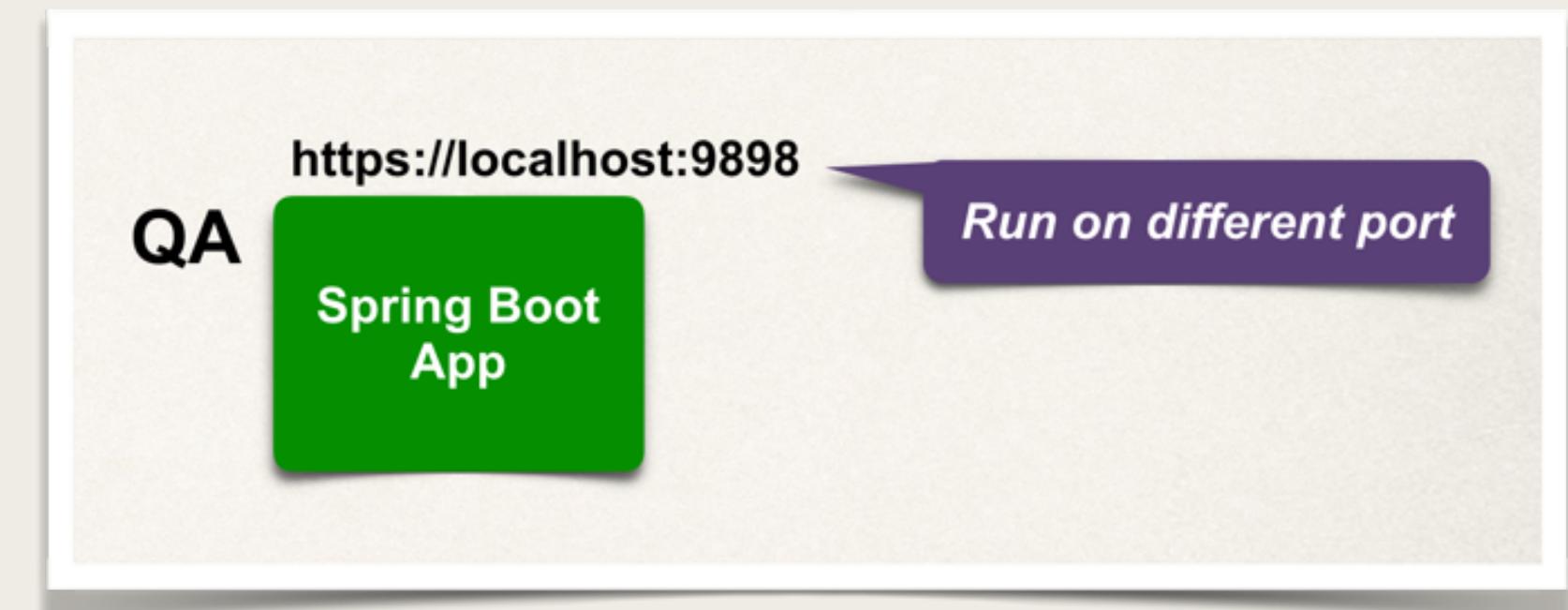
Step 1: Create environment file: environment.qa.ts

environment.qa.ts

```
export const environment = {  
  production: false,  
  
  // add your custom environment configs  
  
  luv2shopApiUrl: "https://localhost:9898/api"  
};
```

Same name

*URL for Spring Boot app
running in QA*



Step 2: Define configuration in angular.json for "build" section

*Path in angular.json file:
projects > angular-commerce > architect > build > configurations*

angular.json

```
...
"configurations": {
  ...
  "qa": {
    "fileReplacements": [
      {
        "replace": "src/environments/environment.ts",
        "with": "src/environments/environment.qa.ts"
      }
    ]
  }
}
```

Configuration name

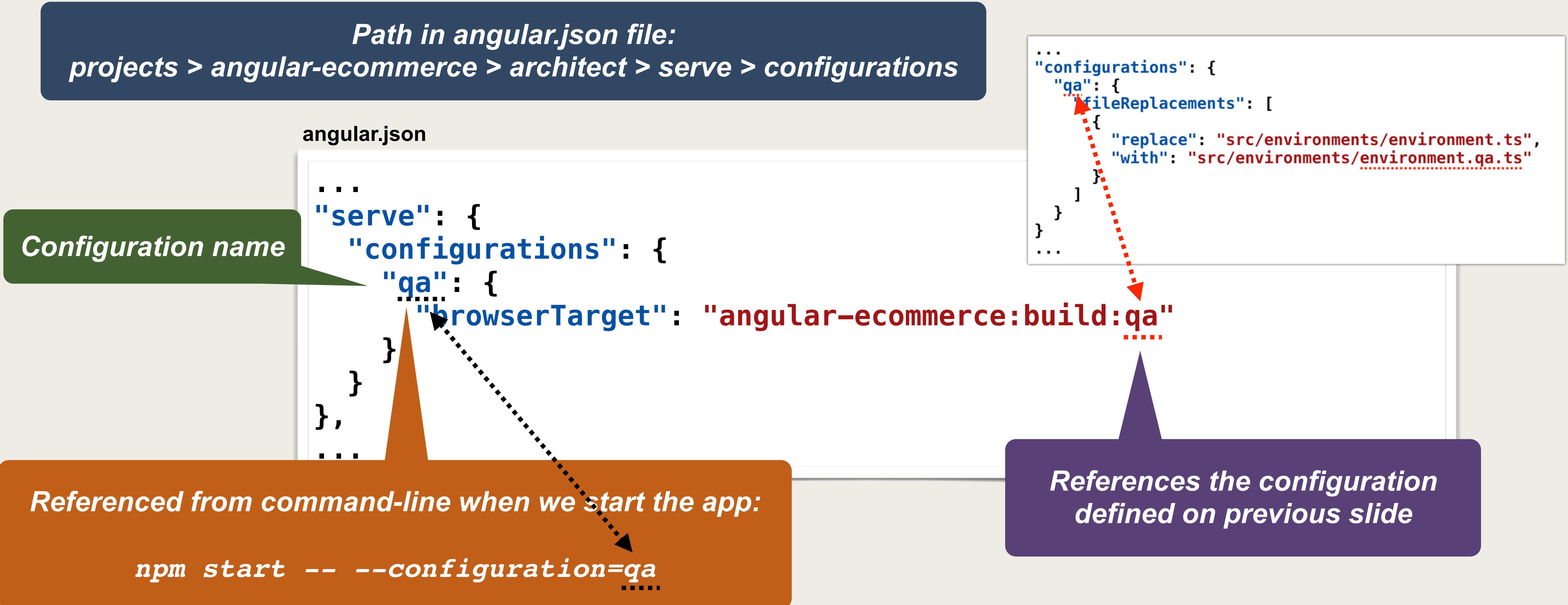
Can use any file name

*Best practice:
Stay consistent with configuration name*

*Makes it easy for other developers to
understand your project*

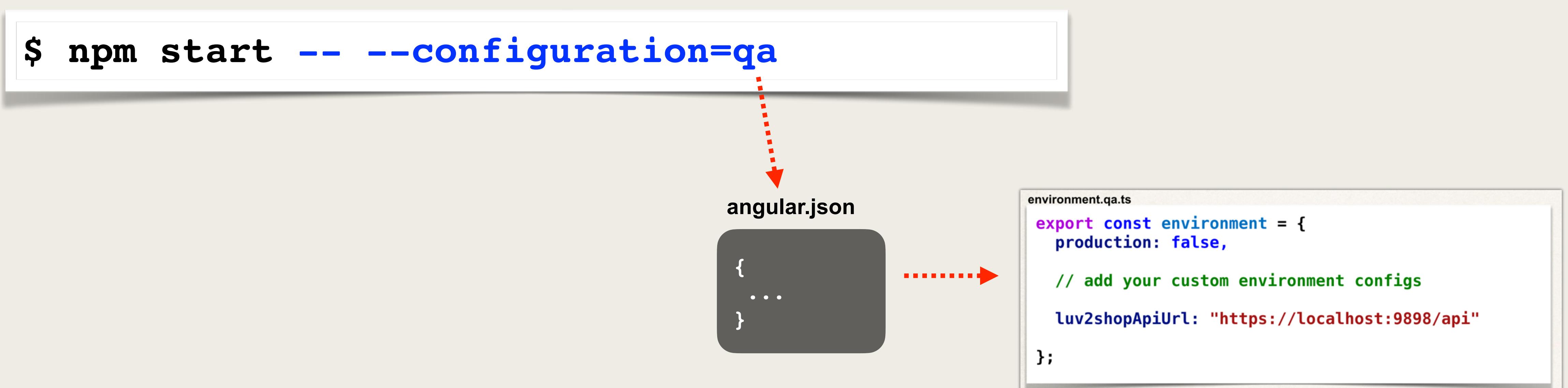
*During processing, for qa configuration,
will load the file: environment.qa.ts*

Step 3: Define configuration in angular.json for "serve" section



Step 4: Run with environment configuration

- Run with qa configuration



Why production: false, production: true???

environment.ts

```
export const environment = {  
  production: false,  
  ...  
};
```

environment.prod.ts

```
export const environment = {  
  production: true,  
  ...  
};
```

main.ts

if true

```
...  
if (environment.production) {  
  enableProdMode();  
}  
...
```

Prod Mode

- Reduce internal logging messages
- Disable file checks for auto-reloading
- Minify scripts and styles
- Eliminate dead-code
- Inline critical CSS and fonts
-

Additional Resources

www.luv2code.com/angular-optimization-configuration