

Chapter 1 - Introduction to Single Page Applications

Objectives

Key objectives of this chapter

- What is a Single Page Application (SPA)?
- Advantages of SPA
- Downsides of SPA
- How to build SPA's using Angular

1.1 What is a Single Page Application (SPA)

Single Page Applications:

- Make a distinction between "pages" and "views".
- Load a single HTML document from the server.
- Use JavaScript and JavaScript libraries to generate views locally by displaying and hiding HTML elements
- Use JavaScript to upload/retrieve data to/from the server.
- Use HTML5 features:
 - ◇ History interface for navigation
 - ◇ Local storage to save and reuse data.
- Are easier to implement using Angular.

Notes

An SPA loads a single HTML document from the server and then selectively shows different DOM elements to create the illusion of multiple page navigation.

A page is a DOM document that is loaded only once. We can show multiple views using the same page.

HTML5 provides the necessary APIs, such as history and local storage, to make SPA possible.

Frameworks like Angular leverage HTML5 and simplify the task of developing complex SPA.

1.2 SPA Workflow

- Example workflow for a single page application:
 - ◇ User clicks a link to open an online email application
 - ◇ Browser retrieves and loads the initial HTML page
 - ◇ Initial view is a list of emails
 - ◇ User clicks an email and view is changed to show details of that email.
 - ◇ User clicks the back button and email list view is displayed again.
- The app did not need to load additional HTML pages for the details view or to return to the list view.

Notes

Example: An email application first shows a list of items in the inbox. When user clicks on an item, the entire screen is updated to show the details of the email thread. When user clicks the back button of the browser the initial list of items is shown again. The transition of pages appear to be identical to any traditional application. Except, all of this is achieved using a single DOM document.

A small to medium size SPA will load all the necessary HTML and JavaScript one time. It will interact with the server by issuing Ajax calls.

A more advanced application can speed up initial load time by only loading the JavaScript and HTML necessary at that time. Subsequently, as new views need to be loaded, it can download additional HTML and scripts using Ajax.

1.3 Traditional Web Application Capabilities

- With traditional web applications users can:
 - ◇ Navigate to new pages using anchor links.
 - ◇ Navigate to previous pages using the back button.

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

- ◇ Return to a pages using bookmarks.
- ◇ Navigation to specific pages by manually entering URLs into the address bar.
- The above capabilities:
 - ◇ Are supported by all browsers.
 - ◇ All rely on downloading separate pages for each view.
 - ◇ Can be achieved in SPA applications using Angular features.

Notes

You have probably developed applications before that show/hide small number of DOM elements in a page. For an SPA however, we change the look of the entire page. This gives the user an impression that she is leaving the old page and moving to a new page. In reality, there is only one page or DOM document that never gets reloaded.

A good SPA should allow the user to hit the back button to go back to the previous state of the page. This requires the use of the HTML5 history API. The application is now responsible for managing the navigation state. In a traditional web application the browser did that for us.

Users should be able to bookmark views. For example, if the user is looking at an email thread she should be able to bookmark it. If user pulls up the page using the bookmark, the application should directly open the email thread view.

SPA's preserve some of the familiar usage paradigm of a regular web application like navigation and bookmarking but brings a few new advantages to the table.

1.4 Single Page Application Advantages

- Navigation between views in SPA's:
 - ◇ Does not require the browser to reload the HTML page
 - ◇ Is more fluid (screen does not flash)

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

- ◇ Is faster
- View data can be:
 - ◇ Downloaded once
 - ◇ Reused in multiple views

Notes

View transition and overall performance are enhanced in SPA's.

The biggest reason behind SPA is rapid page transition. SPA lets user navigate through the application views more rapidly than possible if the application was reloading every page from the server. The transitions also appear visually more fluid. Overall, this creates higher user satisfaction level.

The need for SPA becomes even more urgent when the pages need to load large amounts of data and the backend is slow. For example, a retail web site that lets user filter products based on price, color, gender etc. needs to download the entire catalog. This can be slow, especially when the backend is legacy and slow. SPA lets you download this data once and then share it between different views.

1.5 SPA and Traditional Web Sites

SPA not an all or nothing proposition:

- Large traditional site may include SPAs for specific features:
 - ◇ Mortgage calculator on banking site.
 - ◇ Login/User account maintenance.
 - ◇ Product lookup/display on retail site
- Small applications work well as SPA's
- Large or complex applications may work better as a traditional site or mixed traditional/SPA site

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

Notes:

A "Traditional Web Site" is one where separate pages served by a web server. This would include sites that rely on pages being generated on the server-side using technologies such as Java ServerPages (JSP), Active Server Pages (ASP), PHP, Python, Ruby and others.

Building a small application using a single page is feasible and should definitely be considered. But, building a large and complex application as a single page can become very complicated even using a framework like AngularJS. This is certainly technically possible as Google's Gmail has shown us. But be mindful of the extra man hours and engineering difficulties that this may entail before deciding to go that way.

1.6 SPA Challenges

Challenges that emerge when using SPA's include:

- Search Engine ranking
- Web Application Analytics
- Forward and Back navigation

Notes:

Most of the content of SPA is loaded using Ajax. Most search engines will not be able to index this data. For example if a retail site loads product data as JSON then the page will have a very poor SEO ranking.

Many pages use analytics API to keep track of user activities. Most of these APIs do the tracking at page load time. In an SPA the page is loaded only once. As the user navigates between the views the analytics tool may not be able to capture the transitions.

As the user moves forward from one view to the next, we need to preserve the state of the previous views so that we can restore them when user starts to navigate back. For example, user searches for product and then clicks on a product from the search result.

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

This shows the product details page. When user clicks the back button of the browser we should show the search result. This can be done by either saving the search result or saving the search term and doing a fresh query.

1.7 Implementing SPA's Using Angular

Options for implementing SPA features in Angular include:

- Use of Angular's visibility control features like `ngSwitch` and `[hidden]` to show and hide multiple views inside a single HTML template.
- Using visibility controls along with separate components for each view.
- Using an Angular component router instead of visibility controls to support navigation and to enable additional features such as back button navigation, bookmarking and manual URL entry.

Notes:

Implementing SPA Using Angular

Angular provides several strategies to build SPA. You need to carefully pick your option. It can be as simple as using the `ng-if` tag. That will work for very simple applications. If the template for the screens become large then you can break them up in separate HTML files and create Angular components to dynamically display the screens. For really complex applications you will need to use an Angular component router.

1.8 Simple SPA Using Visibility Control

- Example uses the `ngSwitch` directive to swap between tags in the current HTML template.



Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

- "next" button navigates between sections by incrementing the "index" variable
selector: 'tagswitch',
template: `

```
<div [ngSwitch]="index" class=block>
  <p *ngSwitchCase="1">Page-1</p>
  <p *ngSwitchCase="2">Page-2</p>
  <p *ngSwitchCase="3">Page-3</p>
  <p *ngSwitchDefault>Default</p>
</div><br>
<input type=button (click)="next()"
value="next">`
```

Notes

The code shown above is a section from a custom Angular component. The ngSwitch directive it uses is covered in detail in the "Structural Directives" chapter.

In this case we are only switching between paragraph, <p> tags but the same code could be used to switch between multiple div <div> tags. The content of each div can be whatever is needed by the application. One div could hold a list while another shows a detail section with pictures and another could display avigation elements.

Full Example Component:

```
import { Component } from '@angular/core';

@Component({
  selector: 'tagswitch',
  template: `
<div [ngSwitch]="index" class=block>
  <p *ngSwitchCase="1">Page-1</p>
  <p *ngSwitchCase="2">Page-2</p>
  <p *ngSwitchCase="3">Page-3</p>
  <p *ngSwitchDefault>Default</p>
</div><br>
<input type=button (click)="next()" value="next">
`
})
```

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

```
export class TagSwitch {  
  index: number = 1;  
  next() {  
    this.index = (this.index < 3)? this.index + 1: 1;  
  }  
}
```

1.9 SPA Using Angular Components

- For complex views it is useful to separate the HTML code into separate files.
- This can be done using Angular components.
- The component's selector is used to include the code being switched.

```
<pageone></pageone>
```

- A button or link tag is used to drive the navigation

Page One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas aliquam neque eget quam convallis efficitur. Suspendisse congue, dui placerat aliquam semper, justo quam feugiat diam, eu blandit ex nisl sit amet mi.

next

Notes:

A high level overview of this technique is covered in the next few slides.

The details of creating and using Angular components is covered in the "Components" chapter.

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.10 SPA with Angular Components - Switching

- This Code switches between Angular components:

```
<div [ngSwitch]="index" class=block>
  <pageone *ngSwitchCase="1"></pageone>
  <pagetwo *ngSwitchCase="2"></pagetwo>
  <pagethree *ngSwitchCase="3"></pagethree>
  <p *ngSwitchDefault>Default</p>
</div>
```

- "pageone", "pagetwo" and "pagethree" are the selectors for the components being switched.

Notes

Full example code:

```
import { Component } from '@angular/core';
import { PageOne } from './pageone';
import { PageTwo } from './pagetwo';
import { PageThree } from './pagethree';

@Component({
  selector: 'pageswitch',
  template: `<div [ngSwitch]="index" class=block>
    <pageone *ngSwitchCase="1"></pageone>
    <pagetwo *ngSwitchCase="2"></pagetwo>
    <pagethree *ngSwitchCase="3"></pagethree>
    <p *ngSwitchDefault>Default</p>
  </div>
  <input type=button (click)="next()" value="next">`,
  directives: [PageOne, PageTwo, PageThree]
})

export class PageSwitch {
  index: number = 1;
  next(){this.index = (this.index < 3)? this.index + 1: 1;}
}
```

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.11 SPA with Angular Components - The Displayed Component

- The "pageone" component.("pageone.ts")

```
import { Component } from '@angular/core';
@Component({
  selector: 'pageone',
  templateUrl: './app/pageone.html',
  styles: [':host p {background-color:yellow;font-family:sans-serif;}'],
})
export class PageOne {}
```

- The "pageone.html" file:

```
<h3>Page One</h3><p>
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Maecenas aliquam neque eget quam convallis
efficitur. Suspendisse congue.</p>
```

Notes:

The pageone component uses its templateUrl property to point to an external HTML file.

The external "pageone.html" file includes HTML content that is displayed by the "pageone" component.

Separate components are created for each view being switched.

1.12 Implement SPA Using an Angular Component Router

- Angular Component Routers can be used to switch between views based on the content of the current URL in a browser's address bar.
- The Component Router enables developers to implement:
 - ◇ Forward navigation via links and buttons.

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com

- ◇ Back navigation via the browser's back button.
- ◇ Navigation via bookmarks.
- ◇ Manual navigation by typing into the address bar.
- These features are covered in detail in the Angular Component Router chapter.

1.13 Summary

- In this chapter we covered:
 - The Concept of Single Page Applications (SPA).
 - SPA Advantages.
 - SPA Challenges.
 - Implementation of SPA's using Angular

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

744 Yorkway Place
Jenkintown, PA. 19046
1 877 517 6540
getinfousa@webagesolutions.com