

مستند

معرفی منابع آموزشی

راهنمای آموزشی برنامه نویسان وب

| | | | |
|-------------------------------|---------------------------------|------------|--------------|
| 1396/10/25 | | | تاریخ |
| نسخه 1.0 | | | شماره نسخه |
| فاطمه طاهر خانی و نوید سلاجقه | | | تهیه کنندگان |
| جدول تغییرات: | | | |
| | | | |
| شرح تغییرات | تهیه کنندگان | شماره نسخه | تاریخ |
| تهیه فایل | فاطمه طاهر خانی | 1.1 | 94/05/31 |
| تکمیل | نوید سلاجقه | 2.0 | 96/10/25 |
| تکمیل | مژگان ضیغمی علی تاجمیر ریاحی | 3.0 | 03/03/29 |



فهرست مطالب

| | |
|---|--------------------------|
| 3 | طرح مسئله |
| 4 | منابع یادگیری JavaScript |
| 5 | منابع یادگیری Extjs |
| 8 | در تدکار |
| 9 | چک لیست |

طرح مسئله

به عنوان اولین قدم بهتر است از یک مثال شروع کنیم، فرض کنید می خواهیم صفحه ای شامل یک جدول برای نمایش اطلاعات مشتریان (نام، نام خانوادگی، استان و شهر) داشته باشیم. در این جدول باید امکان اضافه، ویرایش و حذف یک مشتری وجود داشته باشد. (در فاز اول پیاده سازی نیاز نیست کمبو های استان و شهر وابسته باشند)

در این مثال باید این موارد رعایت شود و اطلاعات کافی درباره هر یک ضروری می باشد:

(لازم به ذکر است به هیچ وجه از مدل MVC استفاده نشود و حتما از MVVM استفاده شود)

- 1- Layering (Data,Business,Service,Web)
- 2- Sql Server
- 3- EF Code First
- 4- Repository Pattern
- 5- REST- Web API
- 6- Extjs grid (different model for create and showing data in grid)
- 7- Validation (include server side and client side)
- 8- RTL
- 9- Validation(client-server)
- 10-MVVM Pattern
- 11-Cascading Combo
- 12-Show Client/Server Error

ساختار پروژه به این صورت است که لایه دیتا جهت تعریف Entity ها و DbContext و در لایه بیزینس باید ریپازیتوری ها را تعریف کنیم. در لایه سرویس تمام Api ها و ویو مدل های مورد نیاز تعریف می شوند، در لایه web باید Extjs و اسکریپت های مورد نیاز را قرار دهیم.

استفاده از AutoFac جهت dependency Injection و AutoMapper جهت مپ کردن کلاس ها امتیاز

محسوب میشود.

منابع یادگیری JavaScript

- برای درک بهتر و نحوه درست استفاده از فریم ورک Sencha بهتر است به برخی از مفاهیم و تعاریف در JavaScript هم مسلط باشید. به این منظور مقدمه مناسبی برای شروع در آدرس زیر وجود دارد:

<https://www.w3schools.com/js/default.asp>

- و این لینک نیز می تواند برای آموزش مفید باشد

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript

- برای یادگیری و آشنایی با الگوهای طراحی در JavaScript منبع زیر بسیار مناسب است:

<https://addyosmani.com/resources/essentialjsdesignpatterns/book/#constructorpatternjavascript>

از بین الگوهای تعریف شده در این کتاب ، تسلط به الگوهای زیر در اولویت بیشتری قرار دارد:

- 1- Constructor Pattern
- 2- Module Pattern
- 3- Prototype Pattern
- 4- Mixin Pattern
- 5- MVVM Pattern

انتظار می‌رود پس از مطالعه موارد ذکر شده به مفاهیم زیر آشنایی داشته باشیم:

- 1- JavaScript Prototype
- 2- JavaScript Closure
- 3- Module Pattern

مطالعه این مقاله در درک بهتر مورد 2 و 3 مفید می‌باشد:

<http://www.adequatelygood.com/JavaScript-Module-Pattern-In-Depth.html>

منابع یادگیری Extjs

برای طراحی صفحات از فریم ورک Extjs استفاده می‌کنیم. منبع اصلی یادگیری Extjs سایت Sencha می‌باشد به آدرس: (نسخه 6.5.0)

<https://docs.sencha.com/extjs/6.5.0/index.html>

(در صورت کند بودن از فیلتر شکن استفاده کنید)

در شرکت تدکار و در حال حاضر از Sencha cmd استفاده نمی‌شود.

از فهرست موجود بهتراست این جستارها خوانده شوند:

- 1- Introduction
- 2- Getting started
- 3- 2.Core Concepts
 - i. (همه زیر بخش ها)
- 4- Components
 - i. (همه زیر بخش ها)
- 5- 4. Application Architecture
 - a. Application Architecture
 - b. View Model and Data Binding
- 6- 5. Best Practices

مثال های sencha در این مسیر قرار دارد:

<https://examples.sencha.com/extjs/6.5.0/examples/>

که بسیار مفید و پر کاربرد است و قابل استفاده به صورت Offline می باشد.

برخی نکات که بهتر است در مثال رعایت شوند:

در کد جاوا اسکریپت برای استفاده از هیچ کلاسی از Parent استفاده نشود.
در Extjs برخلاف C# اول constructor کلاس فراخوانی می شود و بعد سازنده پدر . پس در استفاده از متدهای
up و down در initComponents باید دقت شود. از این متدها بیشتر در event استفاده می شود.
در initComponents تا قبل از فراخوانی callParent هنوز کامپوننتها ساخته نشده اند.
Store ها Id نداشته باشند و برای کامپوننت ها از itemId استفاده شود.

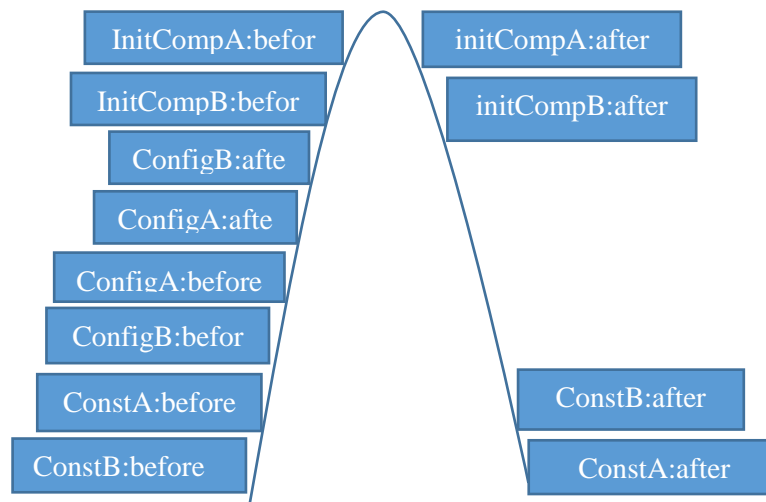
Lifecycle یکی از مباحث مهم در ExtJS می باشد که لازم است اطلاعات کاملی درباره این موضوع داشته باشیم به عنوان مثال می توان دو کلاس تعریف کرد کلاس A و کلاس B ، که کلاس B درون کلاس A می باشد حال موارد زیر را بررسی کرد:

1. initComponents

2. initConfig

3. Constructor

شکل زیر درک درستی از Lifecycle در Extjs به ما میدهد ، در این نمودار کلاسهای A و B بالا نوشته شده اند و در هر کدام initComponent, initConfig, Constructor به همراه callParent پیاده سازی شده اند . در شکل زیر منظور از before قبل از فراخوانی تابع callParent و after بعد از فراخوانی این تابع است.



ابزار Sencha Architect نیز برای طراحی صفحات مورد استفاده است و در مسیر زیر قابل دانلود می باشد.

<https://www.sencha.com/products/architect/#overview>

از این ابزار استفاده کنید و سعی کنید مفهوم Layout را درک کنید ، Layout های مختلف را پیاده سازی کنید و نتایج را بررسی نمایید چراکه این موضوع نیز از مباحث کارا و مهم در ExtJs می باشد.

http://docs.sencha.com/extjs/6.5.3/guides/core_concepts/layouts.html

در تدکار:

در نرم افزار تحت وب شهاب تدکار در حال حاضر از نسخه Extjs 6.2.1 استفاده می شود و الگوی معماری مورد استفاده MVVM می باشد. در این راستا مثال مناسبی وجود دارد که در مسیر زیر قابل مشاهده است:

<http://extjs.eu/example-complex-data-binding-5>

چک لیست

| | |
|---|--|
| Project Structure | |
| Web API | |
| MVVM | |
| RTL | |
| Validation Client | |
| Validation Server | |
| EF | |
| Ext js Layout | |
| Flex | |
| JavaScript ProtoType | |
| JavaScript Closure | |
| Show Server Error | |
| Module Pattern | |
| Inheritance | |
| X__proto__ | |
| InitComponent | |
| initConfig | |
| Constructor | |
| LifeCycle | |
| تعریف توابع خارج از initComponents باشد | |
| تعریف ارایه در config یک کلاس | |
| تعریف فایل های جاوا اسکریپتی جدا ، جدا باشد | |
| تعریف کامپوننت مجزا برای کامبو استان - شهر | |
| توابع getter و setter برای کانفیگ کلاسهای ExtJs | |