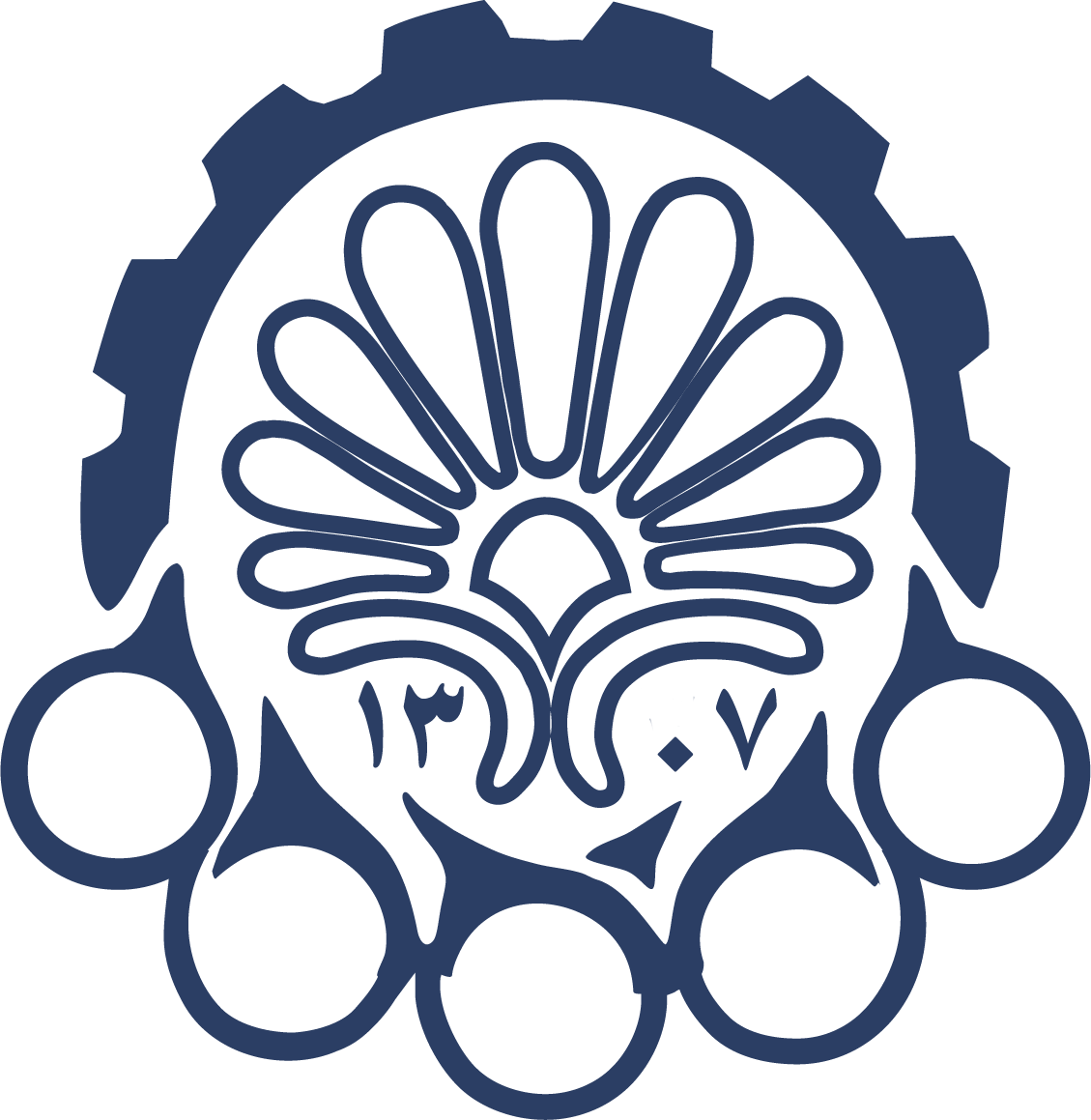
**به نام خدا**



**دانشگاه صنعتی امیرکبیر ( پلی تکنیک تهران )**

دانشکده مهندسی کامپیوتر

**گزارش کارآموزی**

محل کارآموزی : **شرکت مشاوران نرم افزاری اعوان**

استاد :

مهدی نیک نژاد

9630039

تابستان 99

**فهرست مطالب**

[**مقدمه 3**](#_Toc47944615)

[**گزارش دوره کارآموزی 3**](#_Toc47944616)

[**آموزش و فراگیری چند مبحث اولیه 3**](#_Toc47944617)

[**تمرین اول کارآموزی 3**](#_Toc47944618)

[**گزارش ها 5**](#_Toc47944619)

[**شماره گزارش : 1 5**](#_Toc47944620)

[**شماره گزراش : 2 5**](#_Toc47944621)

مقدمه

گزارش دوره کارآموزی

در این قسمت گزارش های دوره کارآموزی اینجانب در شرکت اعوان نوشته شده است . کارآموزی شامل بخش های زیر میباشد :

آموزش و فراگیری چند مبحث اولیه

* + مبحث بازآرایی [[1]](#footnote-1)
  + مبحث آزمون واحد[[2]](#footnote-2)
  + Git
  + کتاب Bill Burke - RESTful Java with JAX-RS 2.0
  + کتاب Robert C. Martin - The Clean Coder\_ A Code of Conduct for Professional Programmers

تمرین اول کارآموزی

مرحله 1

* پیشنیاز

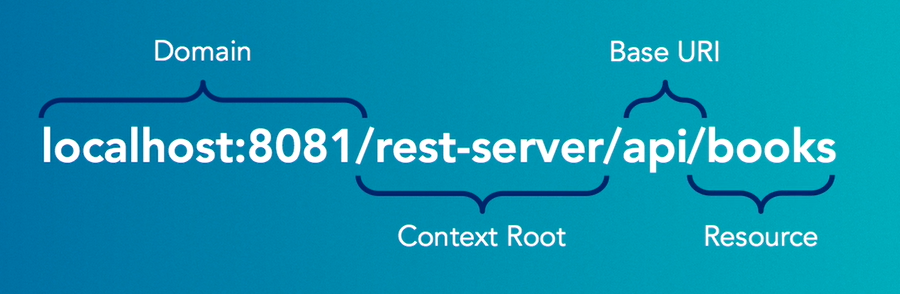
در ابتدای کار باید چند پیشنیاز را داشته باشیم :

* + - نصب jdk8 ( من در ابتدا روی سیستمم jdk11 داشتم که این باعث مشکلاتی شده بود و نسخه 8 را نصب و چایگزین کردم و مشکل حل شد . علت هم آن بود که Spring میخواهد از یکسری کلاس استفاده کند که در جاوا نسخه 11 همگی آنها integrate شده اند و مستقل نمیباشند در صورتی که باید مستقل میبودند برای استفاده . (دقیق نیست!) و همچنین تنظیم JAVA\_HOME برای این نسخه جدید .
    - نصب intellij – قبلا نصب شده بود .
    - پلاگین Lombok را در intellij نصب میکنیم :

Ctrl+alt+S -> Plugins -> search “Lombok” in Marketplace -> install

* + ساخت پروژه
    - با استفاده از [Initializer](http://start.spring.io/) یک پروژه Spring bootساختیم . و تنظیمات اولیه و مورد نیاز را اعمال کردیم و پروژه را ایجاد کردیم و در نهایت آنرا با نرم افزار intellij ، باز کردیم.
  + اجرای سرویس REST
    - در داخل package عه config ، کلاس JerseyConfig را ایجاد کردیم برای ست کردن base URI . که در پروژه های ما معمولا “api” هست .
    - در داخل package عه service ، کلاس GrretingService را ایجاد کردیم برای ست کردن context-root و Resource ها .

مثال :



* + در مورد @Component **:**

کلاس GreetingService را باید تبدیل به Bean کنیم . برای همین باید روی کلاس های GreetingService و JerseyConfig ، انوتیشن @component بگذاریم تا به اصطلاح به کلاس GreetingApplication وصل شوند و موقع اجرا دخیل باشند .

گزارش ها

شماره گزارش : 1

**تاریخ** : 08/05/99 لغایت 17/05/99

توضیح مختصری از آموزش های دیده شده این دوره :

.... بعدا ...

**اقدامات انجام شده :**

فیلم ها و کتاب ها و بقیه منابع شامل لینک ها به مستندات مطالعه و بررسی شدند .

**اقدامات آینده :**

شروع دوره کارآموزی و انجام پروژه های کوچک جهت آشنایی با مفوم REST و تکنولوژی های مرتبط

**مشکلات پیش آمده :**

تنها مشکلی داشتم این بود که میبایست روزد تر بخش آموزش را تمام میکردم اما به دلیل پروژه های دانشگاهی این موضوع کمی طول کشید .

شماره گزراش : 2

**تاریخ** : 17/.05/99 لغایت 19/05/99

توضیح مختصری از آموزش های دیده شده این دوره :

.....

**اقدامات انجام شده :**

کارهایی که در بخش مرحله1 تمرین کارآموزی1 در [docs](https://docs.asta.ir) نوشته شده بود و محول شده بود انجام شدند .

**اقدامات آینده :**

تکمیل مرحله2 از تمرین کاراموزی1

**مشکلات پیش آمده :**

* + در استفاده از git مساله داشتم . برای اینکه جزیی تر بگویم مشکل آنجا بود که repository ای که برای من در گیت شرکت ساخته شده بود ، برنچ master ای داشت که محدودیت برای من گذاشته بود یعنی من میبایست یک برنچ جدید میساختم و روی آن کد میزدم و merge request میفرستادم و فقط منتور بود که میتوانست آن merge request های مرا accept کند تا من ادامه دهم و عملیات push تکمیل شود .
  + در مورد انوتیشن context-root مساله داشتم و نمیدانستم در چه کلاسی باید از آن استفاده کنم که با راهنمایی منتور (آقای احمد حقوقی) متوجه شدم که باید در همان کلاس GreetingService بنویسم .

شماره گزارش : 3

**تاریخ :** 20/.05/99 لغایت 23/05/99

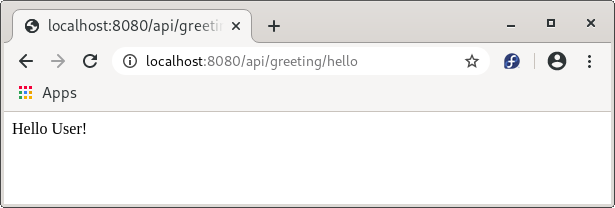
**توضیح مختصری از آموزش های دیده شده این دوره :**

.....

**اقدامات انجام شده :**

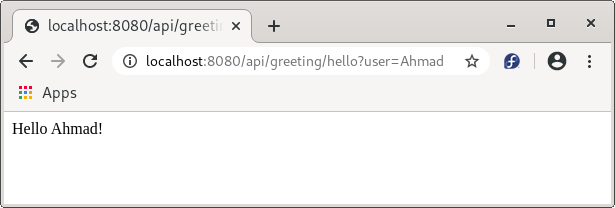
کارهایی که در بخش مرحله2و3 تمرین کارآموزی1 در [docs](https://docs.asta.ir) نوشته شده بود و محول شده بود انجام شدند .

در مرحله قبل ، یک سرویس ساده ساختیم که با نوشتن URL مقابل خروجی موجود در تصویر را داریم :

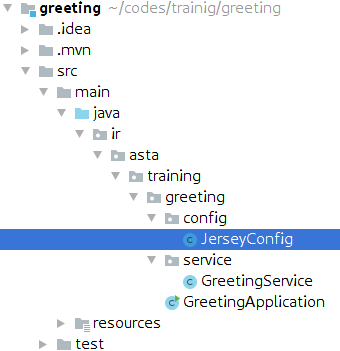


در مرحله 2 آمدیم و Query String را به سرویس اضافه میکنیم . این سرویس ، مقدار را از query string میخواند و مقدار.

Hello را قبل از آن میگذارد



توجه : برای این 2 مرحله و کلا از اینجا به بعد ، کلاس JerseyConfig را در پوشه config ساخته بودیم . این کلاس برای معرفی end-point های Jersey به Spring-boot ایجاد شده است. ساختار سلسله مراتبی به این صورت خواهد بود :



توجه : با استفاده از ApplicationPath می توانید بخش پایه ی URL های پروژه را تنظیم کنید

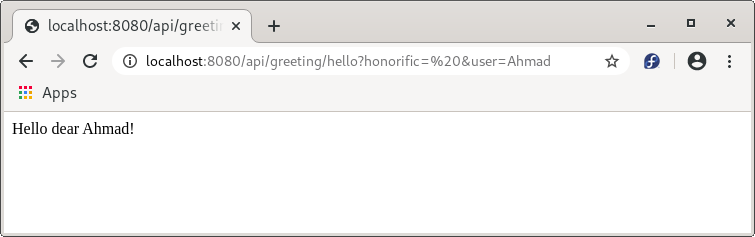
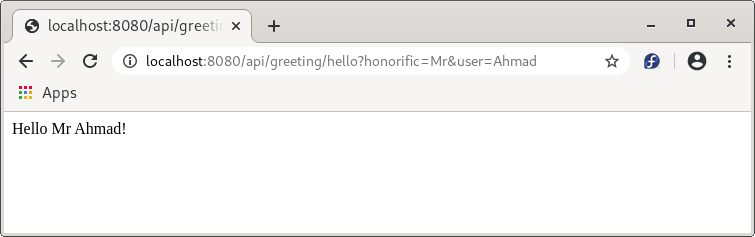
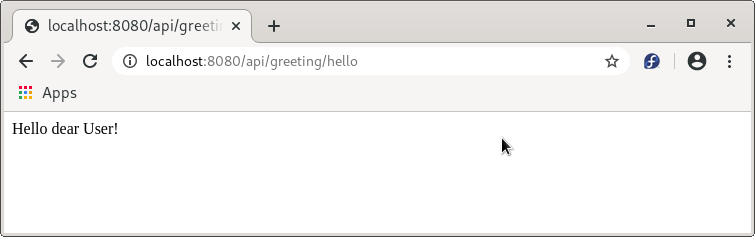
توجه : با استفاده از packages می توانید یک پکیج جاوا را به عنوان محل end-point ها ثبت کنید.

توجه : کلاس سرویس GreetingService را باید تبدیل به Bean کنیم. به این منظور باید روی کلاس Component@ استفاده کنیم.

توجه : اسپرینگ هنگام بالا آمدن وب اپلیکیشن پروژه را اسکن می کند و کلاس هایی که دارای Component@ هستند را Instantiate می کند و هنگام نیاز (مثلا وقتی آدرس یک سرویس را در مرورگر می نویسیم و سرویس را فراخوانی می کنیم.) از آنها استفاده می کند.

حالا در مرحله 3 قرار هست که 2 تا query param بگیریم اولی user و دومی honorific .

تصاویری از خروجی مورد انتظار:



**اقدامات آینده :**

آشنایی با Integration Test و مطالعه مستندات وشاید بازگشت به کتاب Rest

**مشکلات پیش آمده :**

1. مشکل متغیر log در پلاگین Lombok که زیر آن در IDE ، خط قرمز می انداخت . راه حل آن بود که میبایست این پلاگین را نصب میکردم .
2. مشکل در اینکه برای unit test نوشتن چه کلاسی بنویسم ؟ مکان نوشتن را به درستی فهمیده بودم اما اینکه مثلا از فریمورک JerseyTest استفاده کنم یا نه شک داشتم و مستندات هم در این زمینه کامل نبود و در نهایت با کمک منتور (آقای حقوقی) متوجه شدم و کلاس GreetingServiceUnitTest را نوشتم .
3. مشکل دیگر که با گوشزد منتور تصحیح شد در مورد نام متد موجود در کلاس GreetingService بود .که Refactor شد و همچنین مشکل دیگر در پیاده سازی منطق این کلاس بود که پیشنهاد شد به جای پیاده سازی سختی که خودم انجام دادم از متد اماده hasText() در کتابخانه StringUtils استفاده کنم . و همین کار را هم کردم .

شماره گزارش : 4

**تاریخ :** 24/.05/99 لغایت /05/99

**توضیح مختصری از آموزش های دیده شده این دوره :**

.....

**اقدامات انجام شده :**

.......

**اقدامات آینده :**

.......

**مشکلات پیش آمده :**

........

1. Refactoring [↑](#footnote-ref-1)
2. Unit Testing [↑](#footnote-ref-2)