**به نام خدا**

**مستند تولید سانیار**



تهییه ­کننده: مرتضی مهدوی (تیم لید)

تایید کننده: ساناز وحیدی (مالک محصول - طراح سیستم)

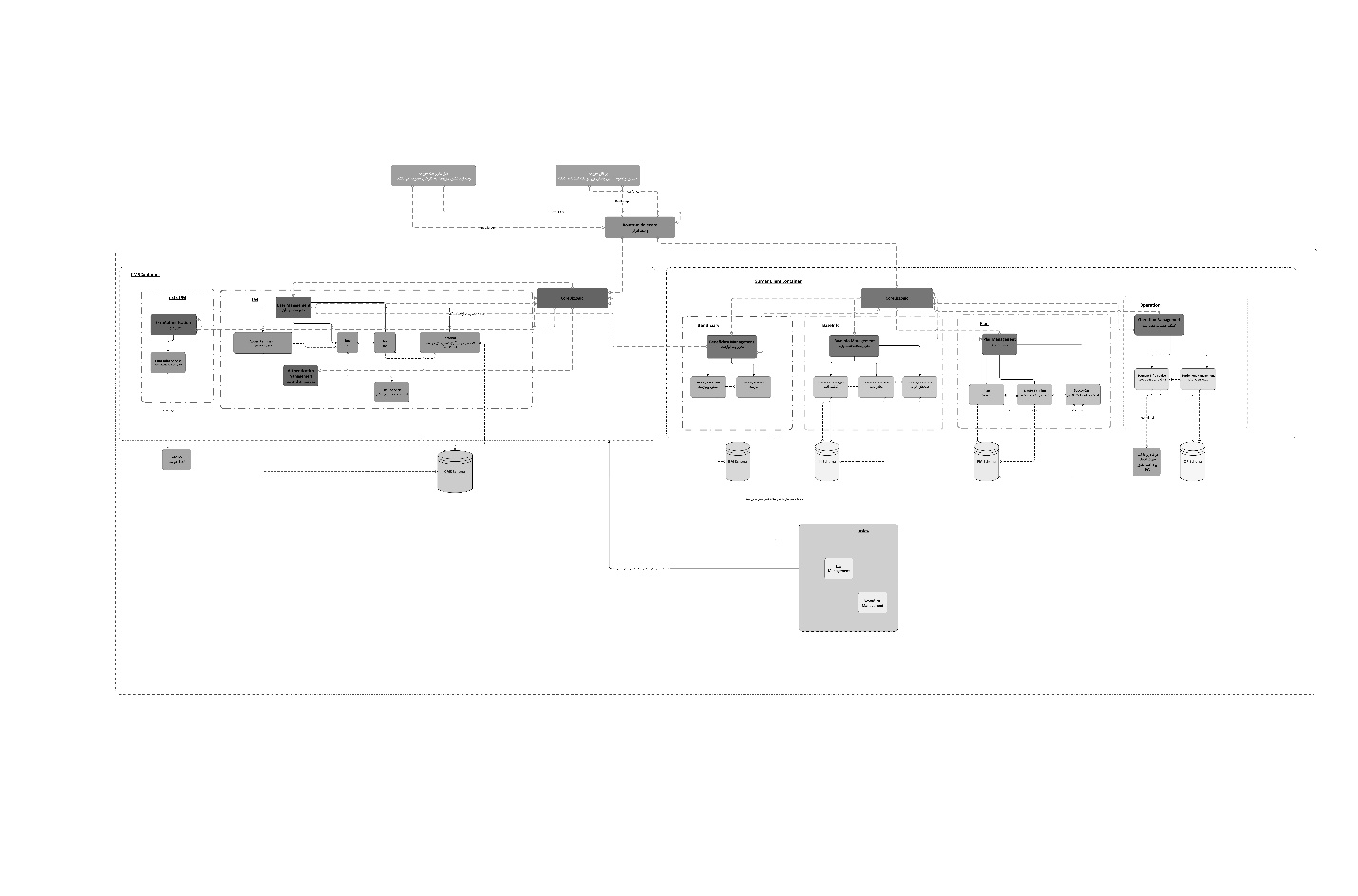
نسخه: 1.0

08/06/1400

**مقدمه**

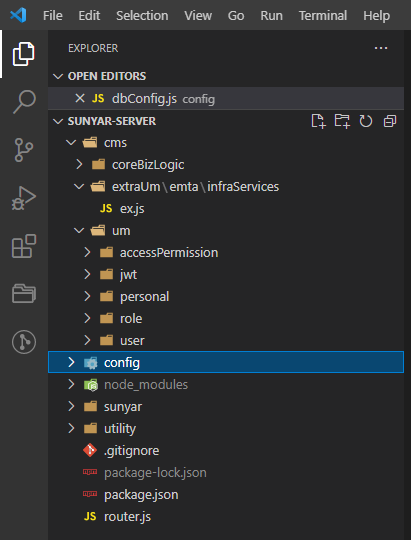
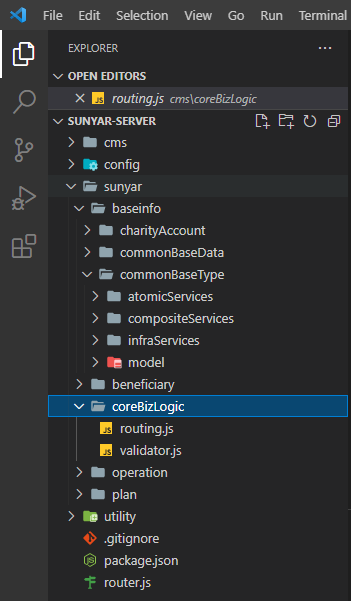
از آنجا که وجود دو سیستم کاملا مستقل در پیاده­سازی نرم افزار نشانگر معماری میکرو سرویس است اما با توجه به مفروضات مطرح شده توسط تیم کسب و کار و معماری کلان ارائه شده (شکل-1) توسط طراح مبتنی بر اینکه زیرسیستم مدیریت محتوا (cms) در آینده قابلیت جداسازی از زیرسیستم سانیار را داشته باشد و از طرفی با درنظر گرفتن محدودیت خیریه­ها در نصب و راه­اندازی سیستم کلی، نمای کلی فولدربندی پروژه جهت برآورده ساختن موارد ذکر شده در زیر مطابق با شکل-2 ارائه گردیده است که در ادامه به تشریح هر بخش به صورت مجزا می پردازیم.

* پیاده سازی به صورت Monolithic
* جابجایی در حد دو سیستم مستقل
* روش توسعه نرم­افزار DDD(Domain Driven Design)
* طراحی ماژولار (مطابق با شکل-1)
* معماری سرویس گرا (مطابق با مستندات پیاده­سازی ارائه شده)

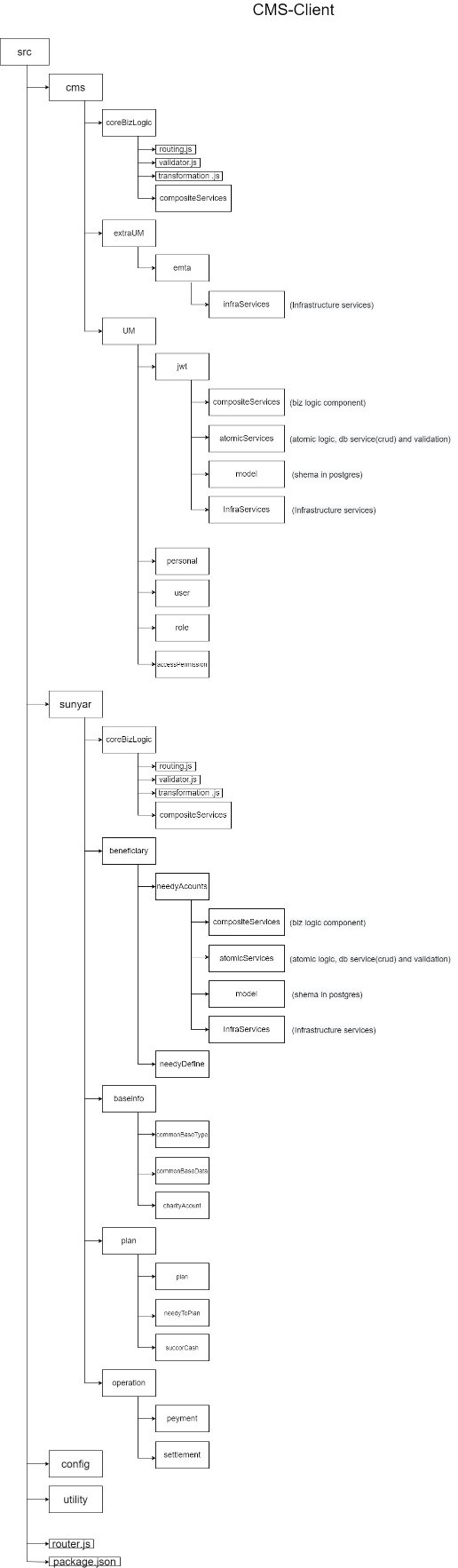


شکل-1-معماری کلان

ملاحظه: همانطور که در شکل 2 نشان داده شده است علاوه بر پوشه cms، sunyar، config، utility، فایل router.js و فایل package.json که به منزله شناسنامه پروژه می باشد به منظور ایجاد یک سرور و هدایت تمامی ریکوست های ارسالی به سمت سرور به گیت وی هر زیرسیستم بر اساس مسیر مشخص شده در ریکوست، هر دو فایل در root اصلی پروژه قرار گرفته اند.

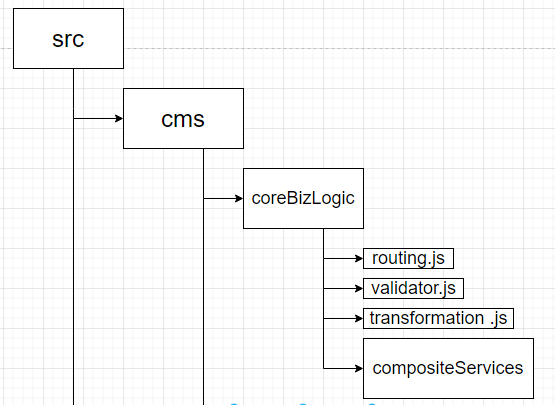
شکل-2-ساختار فولدربندی پروژه در محیط کد – sunyar و cms

****

شکل-3-نمای کلی ساختار فولدربندی

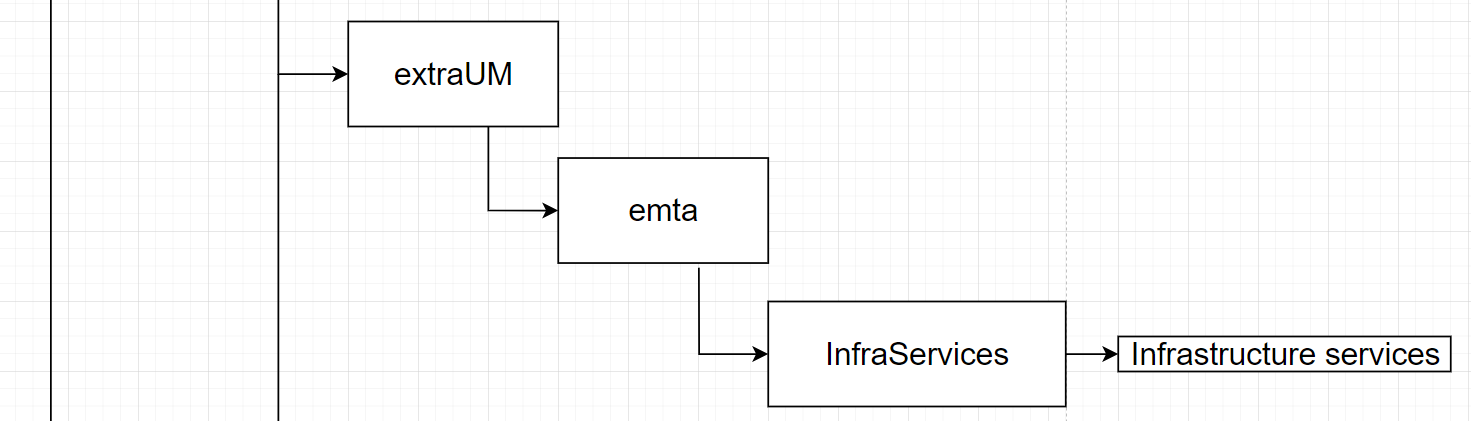
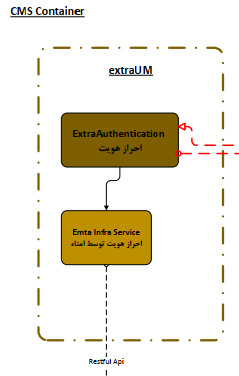
**بخش اول (cms)**

پوشه دامین coreBizLogic از آنجایی که بیزینس خاص خود را دارد نحوه پوشه­بندی داخل آن با بقیه دامین ها متفاوت می باشد. پوشه مذکور دربردارنده فایل routing.js مربوط به زیرسیستم cms است، درون این فایل عملیات authentication و authorization مربوط به پرسنل داخلی خیریه­ها به همراه اعتبارسنجی مقادیر ورودی در سطح اول انجام می­گیرد. پوشه compositeServices وظیفه صدا زدن سرویس­های بیزینسی موجود در باقی دامین­ها را دارد.



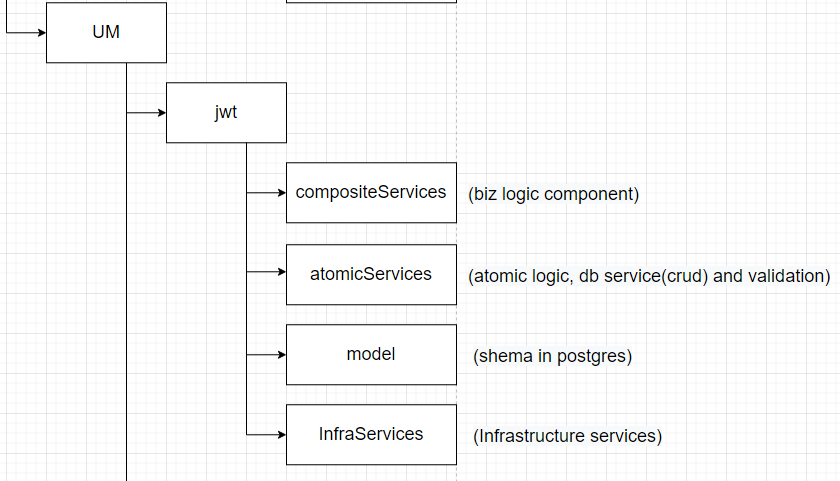
شکل-4-فولدر coreBizLogic زیرسیستم cms

دامین extraUM به منظور احراز هویت بیرونی سیستم تحت پروتکل Oauth توسط سامانه امتا جهت ورود خیرین و در آینده نیازمندان به سیستم درنظر گرفته شده است.



شکل-5-دامین extraUM

در دامین UM نیز کامپوننت jwt به منظور احراز هویت پرسنل خیریه­ها بر اساس jwt قرار داده شده است.

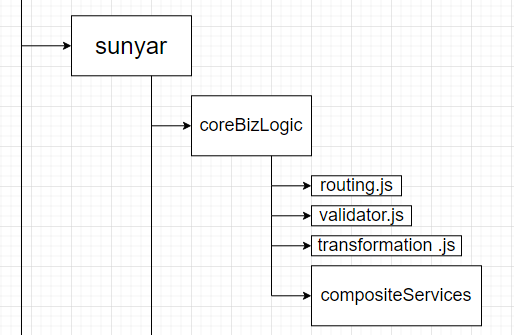


شکل-6-دامین UM

ایجاد دسترسی به سرویس های هر بخش مستلزم دانستن نقش های فرد لاگین شده یا ماهیت او میباشد. در صورتی که فرد ادمین باشد به تمامی سرویس ها دسترسی دارد، اگر نقش مددکار باشد به سرویس های تعریف نیاز مند و تعریف طرح دسترسی دارد، اگر نقش حسابدار باشد به تسویه ها و همچنین اگر ماهیت فرد خیر باشد میتواند طرح هایی که در بخش پورتال نیاز به لاگین دارند دسترسی داشته باشد. عملکرد به این صورت است که توکن jwt که حاوی آیدی کاربر و نام کاربری است، دیکود میشود و اطلاعات نقش ها و ماهیت آن فرد در صورت وجود در دیتابیس برای سرویس های دیگر سرور فرستاده میشود. دسترسی ها میتوانند یک ارایه باشند به عارت دیگر یک سرویس میتواند چندین دسترسی مختلف برای آن تعریف کرد. تمام اطلاعات نقش ها و ماهیت افراد در صورت وجود به پارامتر req.context.auth اضافه میشود که شامل فیلد های roles و personType و user میباشد که استخراج این اطلاعات بر عهده تابع authenticateByJwt که میدلور اصلی است میباشد. تابع authorizeRequest نقش اصلی در تایین دسترسی ها را دارد به این صورت که به عنوان میدلور بر روی هر سرویس قرار میگیرد. ورودی آن اطلاعات فرد لاگین شده و همچنین دسترسی های مجاز برای سرویس است، در صورتی که اطلاعات فرد با این دسترسی های همخوانی نداشته باشد سرور پیغام خطای عدم دسترسی برای درخواست میفرستد. فایل jwt شامل تابع authenticateByJwt و احراز هویت است و فایل

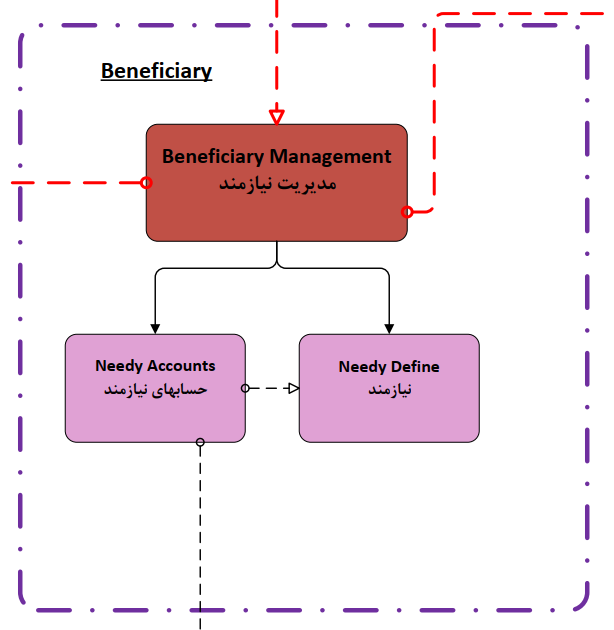
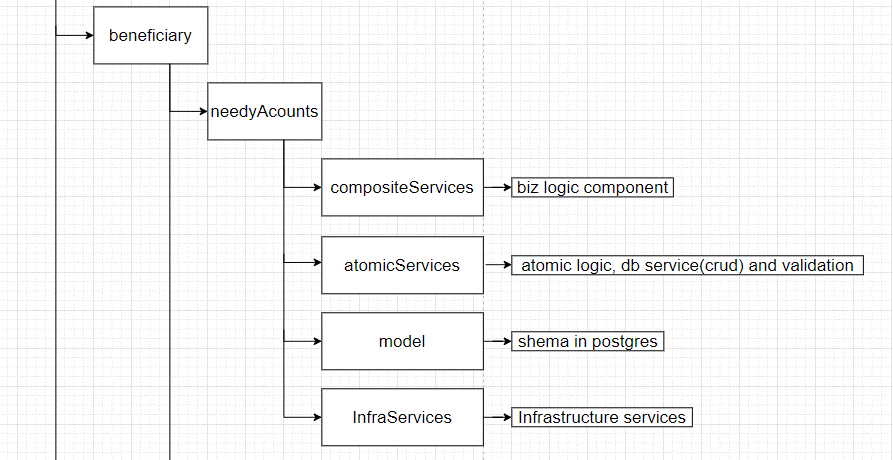
**بخش دوم(sunyar)**

پوشه دامین coreBizLogic از آنجایی که بیزینس خاص خود را دارد نحوه پوشه بندی داخل آن با بقیه دامین ها متفاوت می باشد. پوشه مذکور دربردارنده فایل routing.js مربوط به زیرسیستم سانیار است، درون این فایل اعتبارسنجی مقادیر ورودی در سطح اول انجام می­گیرد و همانند زیرسیستم cms دارای پوشه compositeServices می­باشد.



شکل-7- coreBizLogic سانیار

با توجه معماری کلان پروژه، داخل پوشه sunyar به ازای هر دامین یک پوشه و داخل آن به ازای هر کامپوننت یک پوشه مجزا در نظر گرفته شده است. لایه کامپوننت خود مطابق با شکل 2 به عنوان نمونه که مربوط به دامین beneficiary است شامل چهار بخش مجزا می شود.

شکل-8-دامین beneficiary

**:compositeServices** درون پوشه compositeServices تمامی فرآیندهای مربوط به بیزینس نوشته شده و از سرویسهای crud دیتابیس پیاده سازی شده در پوشه atomicServices استفاده می شود.

**:atomicServices** این پوشه مربوط به کلیه سرویسهای پایه (crud دیتابیس) و همچنین دربرگیرنده ولیدیشن مربوط به بیزینس می باشد.

ملاحظه: سرویسهای پایه به هیچ وجه نباید درون compositeServices قرار داده شوند

**:model** در این پوشه متد تعریف جداول دیتابیسی مربوط به کامپوننت دربردارنده توسط ORM به همراه پارامترهای ورودی فراخوانی می شود.

**:InfraServices** تمامی سرویس های خارجی و زیرساختی اعم از چک سلامت یا بیدار بودن دیتابیس، بالا بودن سرور و ... در این پوشه قرار می گیرند. به طور مثال اگر در کامپوننت مشخص شده در شکل-2، قصد استفاده از سرویسی موجود در cms را داشته باشیم، جای فراخوانی آن درون این پوشه است. این پوشه هیچ ربطی به منطق کسب و کار ندارد.

دسته­بندی مابقی دامین ها و کامپوننت های موجود در زیرسیستم سانیار نیز همانند نمونه آورده شده در بالا می باشند.

**بخش سوم (config)**

فایل های مربوط به پیکربندی دیتابیس به تفکیک Dev (هیچ دیتایی در آن درست نمی باشد) Stage (دیتای عملیاتی و واقعی اما بر روی سرور تست می باشد و صرفا جهت تمرین است) و Operation (همه چیز برای محیط عملیاتی درنظر گرفته می شود)

پیکربندی:Dev کلیه جداول دیتابیس با استفاده از ORM در حالت Drop & Create تعریف می­شوند ( به صورت سیستمی)

پیکربندی :Stageکلیه جداول دیتابیس با استفاده از ORM در حالت چک تعریف می­شوند (به صورت دستی)

پیکربندی :Operationکلیه جداول دیتابیس با استفاده از ORM در حالت چک تعریف می­شوند (به صورت دستی)

**بخش چهارم (utility)**

کلیه فایل هایی که به صورت مشترک به هر دو زیرسیستم سانیار و cms سرویس می دهند و حالت جنرال دارند، مانند لاگ(پیاده سازی در فایل ضمیمه آورده شده است) و توابع پیاده سازی شده جهت انجام ولیدیشن در این پوشه قرار می گیرند.

**مخزن گیت**

آدرس سرور مربوط به مخزن گیت https://192.168.9.71 می­باشد که در آن یک گروه با نام sunyar و سه پروژه با نام­های زیر در داخل آن قرار گرفته است

* sunyar-backend
* sunyar-frontend
* sunyar-core

**کتابخانه­های نصب شده**

**"app-root-path": "^3.0.0",**

**"cookie-parser": "^1.4.5",**

**"crypto-js": "^4.1.1",**

**"express": "^4.17.1",**

**"joi": "^17.4.1",**

**"morgan": "^1.10.0",**

**"nodemon": "^2.0.7",**

**"pg": "^8.7.1",**

**"sequelize": "^6.6.5",**

**"uuid": "^8.3.2",**

**"winston": "^3.3.3"**

**مشخصات زیرساخت**

سرور تست: لینوکسی(CentOS)

* Storage: 60 g
* CPU: 3 core
* RAM: 2 g

سرور دیتابیس: ویندوزی

* Storage: 30 g
* CPU: 5 core
* RAM: 2 g