



دانشگاه تهران  
دانشکده فاریابی  
دانشکده مهندسی  
گروه مهندسی کامپیوتر

## توسعه API تحت وب مدیریت اجرای کنفرانس (با تمرکز بر مدیریت شرکت کنندگان)

نگارش:

مهدی پناهی

استاد راهنما:

دکتر کاظم فولادی قلعه

گزارش پروژه برای دریافت درجه کارشناسی  
در رشته مهندسی کامپیوتر

شهریور ۱۴۰۳



## چکیده

امروزه برگزاری کنفرانس‌های علمی و تخصصی یکی از مهم‌ترین رویدادهای مشارکتی در حوزه‌های مختلف است. سازمان‌دهی و مدیریت این رویدادها چالش‌های زیادی را به همراه دارد که شامل ثبت‌نام شرکت‌کنندگان، برنامه‌ریزی زمان‌بندی ارائه‌ها، دریافت بازخورد، گزارش‌سازی از وضعیت شرکت‌کنندگان و تعامل سازنده با شرکت‌کنندگان است. این پروژه به طراحی و پیاده‌سازی یک سامانه مدیریت کنفرانس می‌پردازد که با هدف ساده‌سازی و بهبود این فرآیندها توسعه یافته است.

سامانه مدیریت کنفرانس توسعه‌یافته در این پروژه دارای ماژول‌های مختلفی است که شامل مدیریت ثبت‌نام شرکت‌کنندگان، برنامه‌ریزی زمان‌بندی کنفرانس و نشست‌های، ایجاد کیوآرکد، ارسال ایمیل‌های اطلاع‌رسانی و نظرسنجی است. این سامانه توسط یک api و کارآمد ارائه می‌شود که امکان استفاده آسان برای سازمان‌دهندگان و شرکت‌کنندگان را فراهم می‌کند.

در طول فرآیند توسعه، نیازها و الزامات سیستم مشخص شده و معماری نرم‌افزاری متناسب با آن طراحی شده است. این سامانه می‌تواند به طور موثری فرآیندهای برگزاری کنفرانس‌ها را بهبود بخشد و به سازمان‌دهندگان و شرکت‌کنندگان در انجام وظایف خود کمک کند.

## فهرست مطالب

۷	فصل اول: زبان و فریمورک
۷	پایتون
۹	جنگو
۱۳	DRF
۱۷	فصل دوم: ابزارها و روش‌های مورد استفاده
۱۷	پایگاه داده SQLite
۱۹	celery
۲۲	احراز هویت JWT
۲۵	signals
۲۷	فصل سوم: ابزارهای کنترل ورژن و تست
۲۷	GIT
۲۷	GitHub
۲۸	PostMan
۲۹	فصل چهارم: ویژگی‌ها، امکانات و نکات پروژه
۲۹	ویژگی‌ها:
۳۰	نکات پروژه:
۳۱	انواع ارسال ایمیل:
۳۲	نحوه ایجاد انواع نظرسنجی:
۳۵	فصل پنجم: api url های
۳۵	Admin panel url
۳۵	Login urls
۳۶	Profile urls
۳۸	Admins urls
۴۱	Events urls
۴۳	Hard delete
۴۴	Participants urls
۴۸	Import / Export file urls
۴۹	Meetings urls
۵۲	Surveys urls

۵۹	..... Qrcode downloader url
۶۰	..... Email logs urls
۶۱	..... خلاصه عملکرد پروژه
۶۳	..... منابع و مراجع
۶۵	..... پیوست‌ها
۶۵	..... لینک گیت‌هاب پروژه



## فصل اول: زبان و فریمورک

### پایتون

یک زبان برنامه‌نویسی سطح بالا، همه کاره و قدرتمند است که به خاطر سادگی، خوانایی و انعطاف‌پذیری‌اش شناخته شده است. این زبان در سال ۱۹۹۱ توسط خیدو ون روسوم طراحی شد و به سرعت به یکی از محبوب‌ترین زبان‌های برنامه‌نویسی در جهان تبدیل شد.

#### مزایای پایتون:

- **سادگی و خوانایی:** پایتون با استفاده از ساختارهای ساده و خوانا، یادگیری و استفاده از آن را برای برنامه‌نویسان مبتدی و با تجربه آسان می‌کند.
- **همه کاره بودن:** پایتون برای طیف وسیعی از کاربردها از جمله توسعه وب، تحلیل داده‌ها، هوش مصنوعی، یادگیری ماشین، پردازش تصویر و اتوماسیون مناسب است.
- **کتابخانه‌های غنی:** پایتون دارای کتابخانه‌های گسترده و قدرتمندی است که به برنامه‌نویسان امکان می‌دهد به سرعت و به طور موثر وظایف مختلف را انجام دهند.
- **جامعه فعال:** پایتون یک جامعه فعال و بزرگ دارد که به برنامه‌نویسان کمک می‌کند تا به سرعت پاسخ سوالات خود را پیدا کنند و از منابع مختلف بهره‌مند شوند.
- **رایگان و متن باز:** پایتون یک زبان برنامه‌نویسی رایگان و متن باز است که به برنامه‌نویسان امکان می‌دهد به طور آزاد آن را دانلود، استفاده و توزیع کنند.

#### کاربردهای پایتون:

- **توسعه وب:** پایتون با فریمورک‌های قدرتمندی مانند Django و Flask، امکان توسعه وب‌سایت‌ها و برنامه‌های کاربردی وب را به طور موثر فراهم می‌کند.
- **تحلیل داده‌ها:** پایتون با کتابخانه‌های قدرتمندی مانند NumPy، SciPy و Pandas، امکان تحلیل داده‌ها، تجسم داده‌ها و یادگیری ماشین را فراهم می‌کند.
- **هوش مصنوعی و یادگیری ماشین:** پایتون به عنوان زبان اصلی در حوزه هوش مصنوعی و یادگیری ماشین شناخته شده است. کتابخانه‌های قدرتمندی مانند TensorFlow، PyTorch و Keras به برنامه‌نویسان امکان می‌دهد تا مدل‌های یادگیری ماشین را پیاده‌سازی و آموزش دهند.
- **پردازش تصویر:** پایتون با کتابخانه‌های قدرتمندی مانند OpenCV و Pillow، امکان پردازش تصویر، تشخیص چهره و شناسایی اشیاء را فراهم می‌کند.
- **اتوماسیون:** پایتون با کتابخانه‌های قدرتمندی مانند Selenium و PyAutoGUI، امکان اتوماسیون وظایف تکراری را فراهم می‌کند.

### معایب پایتون:

- **سرعت:** پایتون به عنوان یک زبان تفسیر شده، می تواند در مقایسه با زبان های کامپایل شده مانند ++C کندتر باشد.
- **مصرف حافظه:** پایتون می تواند به دلیل استفاده از تخصیص حافظه پویا، حافظه بیشتری را نسبت به زبان های دیگر مصرف کند.
- **خطاهای زمان اجرا:** پایتون در زمان اجرا ممکن است خطاهایی را نشان دهد که در زبان های کامپایل شده در زمان کامپایل شناسایی می شوند.
- **محدودیت در برنامه نویسی سیستم:** پایتون به طور سنتی برای برنامه نویسی سیستم مناسب نیست و در مقایسه با زبان های دیگر مانند ++C در این زمینه محدودتر است.

پایتون یک زبان برنامه نویسی قدرتمند و همه کاره است که به خاطر سادگی، خوانایی و انعطاف پذیری اش شناخته شده است. این زبان برای طیف وسیعی از کاربردها مناسب است و به طور فزاینده ای در جهان در حال گسترش است. با این حال، پایتون دارای معایبی مانند سرعت پایین، مصرف حافظه بالا و خطاهای زمان اجرا است که باید در نظر گرفته شوند.



## جنگو

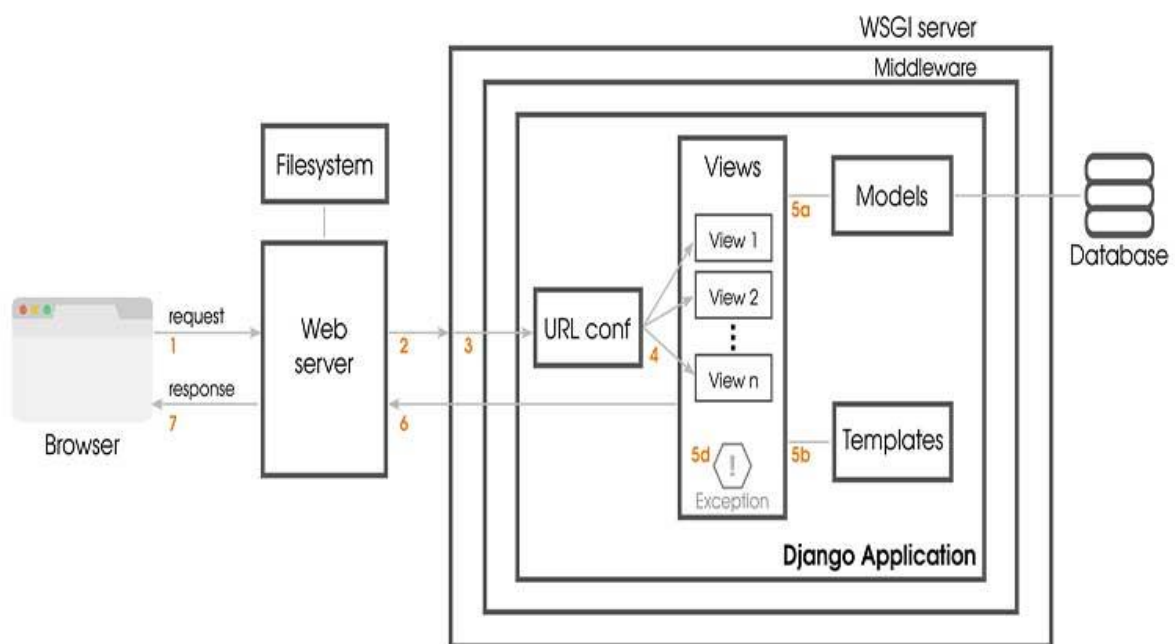
جنگو (Django) در سال ۲۰۰۳ توسط Adrian Holovaty و Simon Willison هنگامی که شروع به استفاده از پایتون برای توسعه یک برنامه وب و طراحی سایت کردند، ایجاد شد. آنها در روزنامه لارنس ژورنال-ورلد کار می‌کردند و به وضوح ایده خوبی از آنچه می‌خواستند بدست بیاورند، داشتند.

سرانجام فریمورک جنگو در ژوئیه ۲۰۰۵ با مجوز BSD منتشر شد و نام آن را بر اساس گیتاریست Django Reinhardt نهادند. در سال ۲۰۰۸، بنیادی به نام بنیاد نرم افزار جنگو (DSF) برای نگهداری و توسعه جنگو تأسیس شد.

### جنگو چیست؟

جنگو یک فریمورک وب مبتنی بر پایتون و زبان سطح بالا می‌باشد که امکان طراحی و توسعه سریع وب سایت‌های ایمن را فراهم می‌کند. جنگو توسط توسعه دهندگان باتجربه ساخته شده و از بسیاری از مشکلات هنگام تولید وب سایت پیشگیری می‌کند. این برنامه رایگان و منبع باز (Open Source) است، همچنین دارای یک انجمن پرکاربرد و فعال، مستندات عالی و گزینه‌های زیادی برای پشتیبانی رایگان و غیر رایگان می‌باشد.

### جنگو چگونه کار می‌کند؟



وقتی درخواستی به یک سرور وب می‌رسد، ابتدا این درخواست به جنگو منتقل می‌شود و جنگو سعی می‌کند بفهمد چه چیزی مورد درخواست واقع شده است. همچنین جنگو آدرس صفحه وب را می‌گیرد و سعی می‌کند بفهمد چه کاری باید انجام دهد. این قسمت توسط Django url resolver انجام می‌شود (توجه داشته باشید که آدرس وب سایت URL نامیده می‌شود که مخفف Uniform Resource Locator می‌باشد). سپس جنگو لیستی از الگوها را می‌گیرد و سعی می‌کند URL را با آنها تطبیق دهد.

جنگو الگوها را از بالا به پایین بررسی می کند و اگر درخواست با الگوی خاصی مطابقت داشته باشد، جنگو درخواست را به عملکرد مرتبط منتقل می کند (که به آن View می گویند).

یک حامل نامه را با یک نامه تصور کنید. او در خیابان قدم می زند و هر شماره خانه را با شماره موجود در نامه چک می کند. اگر مطابقت داشته باشد، نامه را آنجا می گذارد. urlresolver نیز همین گونه کار می کند!

در عملکرد view، همه کارهای زیر انجام میشود: ما می توانیم به یک پایگاه داده متصل شویم تا به دنبال اطلاعات خاصی باشیم. شاید کاربر خواسته است چیزی را در داده تغییر دهد. مانند نامه ای که می گوید: "لطفاً فلان شرح کار را تغییر دهید." همچنین می تواند بررسی کند که آیا شما مجاز به انجام این کار هستید یا نه، سپس توضیحات را برای شما به روز می کند و پیامی را ارسال می کند: "انجام شد!" و جنگو می تواند آن را به مرورگر وب کاربر ارسال کند.

## مزایای فریم ورک جنگو

### ۱. زبان برنامه نویسی کامل

جنگو تقریباً امکان همه کارهایی را که ممکن است توسعه دهندگان بخواهند انجام دهند را فراهم می کند. از آنجا که هر آنچه شما نیاز دارید بخشی از یک "محصول" است، همه اینها به طور یکپارچه با هم کار می کنند و از اصول طراحی سازگار با هم پیروی می کنند و مستندات گسترده و به روز دارند.

### ۲. همه منظوره

از جنگو می توان تقریباً برای ساختن هر نوع وب سایتی استفاده کرد. این برنامه می تواند با هر فریم ورک سمت مشتری دیگری نظیر فریم ورک های php نیز کار کند و می تواند تقریباً در هر قالب (از جمله HTML، RSS، JSON، XML، PHP و غیره) محتوایی را ارائه دهد.

### ۳. پشتیبانی از انواع کامپوننت ها

همچنین جنگو تقریباً برای هر عملکردی که بخواهید گزینه هایی را برای شما فراهم می کند (به عنوان مثال چندین پایگاه داده معروف، موتورهای templating و غیره)، همچنین می توان در صورت لزوم از سایر مولفه های مورد نیاز نیز استفاده کرد.

### ۴. امنیت

جنگو با ارائه یک فریم ورک قدرتمند جهت محافظت خودکار از وب سایت طراحی شده است و به توسعه دهندگان کمک می کند تا از بسیاری از اشتباهات معمول امنیتی جلوگیری کنند. به عنوان مثال، جنگو یک روش ایمن برای مدیریت حساب های کاربری و رمزهای عبور کاربر، اجتناب از اشتباهات رایج مانند قرار دادن اطلاعات جلسه (Sessions) در کوکی ها و در جاهایی که آسیب پذیر میباشد یا ذخیره مستقیم رمزهای عبور به جای هش نمودن آنها و... فراهم می کند.

همچنین جنگو به طور پیش فرض از بسیاری از آسیب پذیری‌ها نیز محافظت می‌کند، از جمله تزریق SQL، برنامه نویسی بین سایت، جعل درخواست بین سایت و کلیک کردن و ... .

## ۵. مقیاس پذیر بودن

جنگو از معماری مبتنی بر مولفه "عدم اشتراک گذاری (shared-nothing)" استفاده می‌کند. بدین معنا که هر قسمت از معماری مستقل از قسمت‌های دیگر است و بنابراین می‌تواند در صورت لزوم جایگزین شده یا تغییر کند.

## ۶. امکان گسترش منابع

وجود یک تفکیک واضح بین قسمت‌های مختلف به این معنی است که می‌تواند با افزودن سخت افزار در هر سطح، افزایش سرورهای ذخیره، سرورهای پایگاه داده یا سرورهای برنامه، امکان افزایش تعداد بازدیدکنندگان و در نتیجه افزایش رتبه سئو سایت را ایجاد کند. بسیاری از سایت‌های پرتراфик از جنگو استفاده کرده اند تا خواسته‌های آنها را برآورده کند (به عنوان مثال Instagram و Disqus و ...).

## ۷. قابلیت ارتقا و نگهداری

کد جنگو با استفاده از اصول و الگوهای طراحی نوشته می‌شود که ایجاد کد قابل نگهداری ساده برای طولانی مدت و قابل استفاده مجدد را فراهم می‌کند. همچنین جنگو از اصول DRY یا Don't Repeat Yourself استفاده می‌کند بنابراین هیچ کپی غیرضروری وجود ندارد، و میزان کدنویسی را کاهش می‌دهد. همچنین جنگو از فابلیت MVC نیز پشتیبانی می‌کند.

## ۸. پشتیبانی از چندین سرور

جنگو به زبان پایتون نوشته شده است و در بسیاری از سیستم عامل‌ها اجرا می‌شود. این بدان معناست که شما به هیچ بستر سرور خاصی متکی نیستید و می‌توانید برنامه‌های خود را بر روی بسیاری از میزبانی‌های وب اجرا کنید.

به علاوه، جنگو توسط بسیاری از ارائه دهندگان خدمات میزبانی وب، که اغلب زیرساخت‌های خاص و مستندات میزبانی سایت‌های جنگو دارند نظیر هاست پایتون نیز ارائه میشود.

## ۹. پنل مدیریت

هنگام نصب برنامه، پنل مدیریت جنگو به طور خودکار نصب می‌شود. این باعث می‌شود که توسعه دهنده نیازی به ایجاد دستی پنل مدیریت نداشته باشد.

با استفاده از برنامه‌های شخص ثالث، کنسول مدیریت پیش فرض جنگو می‌تواند بهبود یافته و متناسب با نیازهای پروژه شما سازگار شود. همچنین، این فریمورک به شما این امکان را می‌دهد که رابط کاربری پنل مدیریت پیش فرض را سفارشی کنید.

#### ۱۰. سئو

کدی که در پایتون نوشته شده حتی برای افراد غیر حرفه ای نیز قابل خواندن و درک است. این یکی از عواملی است که باعث می شود برنامه های وب پایتون، سازگار با مبانی سئو (SEO Friendly) باشند. جنگو URL های معنایی بر اساس عنوان و محتوای صفحه سایت تولید می کند. همچنین برنامه های جنگو سایر ویژگی های بهینه سازی موتور جستجو را نیز به راحتی پیاده سازی می کنند.

## DRF

### رست فریمورک جنگو (Django rest framework) چیست؟

در دنیای فناوری اطلاعات و به ویژه در زمینه توسعه وب، فریمورک های مختلفی وجود دارند که به برنامه نویسان کمک می کنند تا وب سایت ها و اپلیکیشن های پیچیده و کارآمدی بسازند. یکی از این فریمورک ها که به ویژه برای توسعه وب سرویس های RESTful استفاده می شود، رست فریمورک جنگو یا به اختصار (DRF) Django Rest Framework است. در این مقاله ما به معرفی جامع این فریمورک و مزایای استفاده از آن برای توسعه وب سرویس ها می پردازیم امیدواریم این مقاله برای توسعه دهندگان وب و افرادی که به دنبال خرید هاست پایتون هستند مفید باشد.

### معرفی Django Rest Framework (DRF)

یک فریمورک توسعه وب سرویس های RESTful است که بر پایه فریمورک محبوب جنگو ساخته شده است. با استفاده از این فریمورک، برنامه نویسان می توانند به سادگی و سرعت بالا وب سرویس های RESTful پیچیده را ایجاد، مدیریت و ارتباط برقرار کنند. در ادامه، ویژگی ها و مزایای این فریمورک قدرتمند را بیشتر بررسی خواهیم کرد:

#### مدیریت کارآمد مدل ها:

مدیریت کارآمد مدل ها یکی از ویژگی های برجسته Django Rest Framework (DRF) است که این فریم ورک را از دیگر فریم ورک های توسعه وب سرویس های RESTful متمایز می کند. در ادامه، به جزئیات بیشتری در مورد این ویژگی مهم پرداخته خواهیم کرد:

- **انتزاع و انعطاف پذیری:** یکی از مزیت های بزرگ DRF این است که از مدل های Django به عنوان منبع اصلی برای داده ها استفاده می کند. این به برنامه نویسان این امکان را می دهد تا با استفاده از انعطاف پذیری مدل های Django سریع تر و با دقت بیشتری وب سرویس های RESTful را ایجاد کنند. هر تغییری که در مدل ها اعمال شود، به طور خودکار در وب سرویس ها نیز اعمال می شود.
- **مدیریت رابطه ها:** مدل های Django می توانند روابط مختلفی را با یکدیگر داشته باشند (مانند روابط یک به چند و یا روابط چند به چند). DRF به برنامه نویسان این امکان را می دهد تا به سادگی این روابط پیچیده را در وب سرویس های خود نمایش دهند و از آن ها استفاده کنند.
- **اعتبارسنجی داده ها:** یکی از ویژگی های مهم DRF این است که به برنامه نویسان اجازه می دهد تا داده ها را قبل از ذخیره سازی در پایگاه داده اعتبارسنجی کنند. این اعتبارسنجی می تواند شامل اعتبارسنجی انواع داده، اعتبارسنجی ارتباطات و روابط و یا هر گونه قانونی دیگری باشد که برنامه نویس می خواهد اعمال کند.
- **نمایش داده ها:** DRF به برنامه نویسان امکان می دهد تا داده های مدل ها را در فرمت های مختلف مانند JSON یا XML نمایش دهند. این امکان به وب سرویس ها این امکان را می دهد تا با انواع دیگری از سیستم ها و اپلیکیشن ها ارتباط برقرار کنند.

- **مدیریت آپلود فایل :** DRF امکانات خوبی برای مدیریت فایل های آپلود شده در درخواست های وب سرویس ها دارد. این امکان به برنامه نویسان اجازه می دهد تا به سادگی فایل های آپلود شده را دریافت و ذخیره کنند.

به طور کلی، مدیریت کارآمد مدل ها در DRF به برنامه نویسان امکان می دهد تا با استفاده از مدل های Django، وب سرویس های RESTful پیچیده ای را با دقت و سرعت بالا ایجاد و مدیریت کنند. این ویژگی و امکانات جنگو بهره می برد و به برنامه نویسان این اطمینان را می دهد که داده ها به درستی مدیریت و نمایش داده می شوند.

### ساختار قوی درخواست و پاسخ:

ساختار قوی درخواست و پاسخ یکی از ویژگی های برجسته در Django Rest Framework (DRF) است که این فریم ورک را به یکی از پرکاربردترین فریم ورک های توسعه وب سرویس های RESTful تبدیل کرده است. در این ساختار، درخواست هایی که از سمت مشتری ها به وب سرویس ارسال می شوند و پاسخ هایی که از سمت وب سرویس به مشتری ها ارسال می شود، به یک فرمت مشخص و قابل فهم تبدیل می شوند. در ادامه، جوانب مختلف این ساختار قوی را بررسی خواهیم کرد:

- **پارامترهای درخواست :** درخواست های وب سرویس ها شامل اطلاعاتی مانند مسیر (URL، POST، PUT، DELETE و غیره)، پارامترهای درخواست و هدرها (headers) می شوند. DRF با استفاده از این پارامترها تشخیص می دهد که کاربر درخواستی به وب سرویس ارسال کرده است و چه اقداماتی باید انجام شود.

- **نوع محتوای درخواست و پاسخ :** محتوای درخواست و پاسخ می تواند به صورت JSON، XML و یا دیگر فرمت های مشخصی باشد. DRF از این محتواها برای تعامل با داده ها استفاده می کند. معمولاً JSON به عنوان فرمت محتوای استاندارد برای درخواست ها و پاسخ ها در RESTful وب سرویس ها استفاده می شود.

- **نمایش داده ها در پاسخ :** یکی از وظایف اصلی وب سرویس های RESTful نمایش داده ها به دقت و ساختارمند است.

- **کدهای وضعیت HTTP:** وب سرویس ها به طور معمول با کدهای وضعیت HTTP ارتباط با درخواست ها و پاسخ ها را مدیریت می کنند.

- **اعتبارسنجی و امنیت :** درخواست ها باید معتبر و امن باشند. DRF از امکانات اعتبارسنجی (Validation) برای بررسی درخواست ها و از امکانات امنیتی مانند مدیریت CSRF برای جلوگیری از حملات امنیتی استفاده می کند.

- **مدیریت خطاها:** DRF قابلیت های خوبی برای مدیریت و بازگشت خطاها در صورتی که درخواست نامعتبر باشد یا مشکلی در سمت سرور رخ دهد، دارد. این امکان به برنامه نویسان این اطمینان را می دهد که مشتری ها از خطاهای مناسب مطلع می شوند.

به طور کلی، ساختار قوی درخواست و پاسخ در DRF از برنامه نویسان می خواهد که بیشتر درگیر منطق کاربردی خود باشند و مشخص نمایند که چه چیزی باید با درخواست ها انجام شود و چگونه پاسخ ها باید ساخته شوند. این سهولت و انعطاف پذیری در مدیریت درخواست ها و پاسخ ها به برنامه نویسان اجازه می دهد تا وب سرویس های با کیفیت و قابل اعتمادی ارائه دهند.

## مدیریت احراز هویت و دسترسی

مدیریت احراز هویت و دسترسی یکی از مهم ترین ویژگی های Django Rest Framework (DRF) است که به برنامه نویسان اجازه می دهد کنترل دقیقی بر روی کاربران، نقش ها و دسترسی ها در وب سرویس ها اعمال کنند. در اینجا نکات کلیدی در مورد مدیریت احراز هویت و دسترسی در DRF را بیشتر توضیح خواهیم داد:

احراز هویت DRF (Authentication): از انواع مختلفی از احراز هویت پشتیبانی می کند. برخی از روش های معمول احراز هویت که در DRF استفاده می شوند عبارت اند از:

- **Token Authentication:** در این روش، یک توکن مخصوص به هر کاربر ارسال می شود که به ویژگی های اوصاف کاربر (مانند نقش ها) مرتبط است. این توکن در هدرهای درخواست ها قرار می گیرد و به عنوان یک نشانه معتبری از هویت کاربر درخواست را تأیید می کند.
- **Session Authentication:** در این روش، هویت کاربران بر اساس نشست های سروری (sessions) که با استفاده از کوکی ها ایجاد می شوند، مدیریت می شود. این روش معمولاً برای وب سایت هایی که باید وضعیت ورود کاربر را در طول یک نشست نگه دارند، مناسب است.
- **Basic Authentication:** این روش احراز هویت از طریق نام کاربری و رمز عبور کاربر انجام می شود که به صورت Base64 در هدرهای درخواست ارسال می شود. این روش بیشتر برای موارد خاص و نیازمند امنیت بالا استفاده می شود.

• **اجازه ها و نقش ها DRF (Permissions and Roles):** به برنامه نویسان امکان می دهد تا با دقت کنترل کنند که کدام کاربران به چه نوع داده ها و منابعی دسترسی دارند. برای این کار از نقش ها (Roles) و اجازه ها (Permissions) استفاده می شود:

• نقش ها: برنامه نویسان می توانند نقش های مختلفی برای کاربران تعریف کنند. هر نقش می تواند مجموعه ای از اجازه ها را داشته باشد.

• **اجازه ها:** اجازه ها مشخص می کنند که کدام کاربران به چه عملیات ها (مثل خواندن، نوشتن، به روزرسانی و حذف) در داده ها دسترسی دارند. DRF از اجازه های پیش فرض مانند IsAuthenticated و IsAdminUser پشتیبانی می کند و همچنین این امکان را می دهد تا اجازه های سفارشی نیز تعریف شوند.

• **مدیریت خطاها:** DRF اجازه می دهد تا در صورتی که کاربر اجازه دسترسی به یک منبع را نداشته باشد، یا درخواست نامعتبری ارسال کند، پاسخ های مناسب و خطاهای قابل فهمی ارسال شوند. این امکان به برنامه نویس این اطمینان را می دهد که کاربران مطلع به درستی از هر مشکل یا نقصی در درخواست ها هستند.

با توجه به این ویژگی ها، DRF به برنامه نویسان امکان می دهد تا سیستم های احراز هویت پیچیده و دقیقی را برای وب سرویس های RESTful ایجاد کنند و کنترل دقیقی بر روی دسترسی کاربران به منابع داشته باشند. این قابلیت ها به برنامه نویسان این امکان را می دهد تا وب سرویس هایی با امنیت بالا و سازگار با استانداردهای امنیتی را پیاده سازی کنند.

## امنیت بالا

DRF از روش های امنیتی پیشرفته ای برای جلوگیری از حملات مختلف مانند حملات SQL injection و حملات CSRF (Cross Site Request Forgery) استفاده می کند. همچنین، این فریمورک از ویژگی هایی برای اعتبارسنجی داده ها و کنترل دسترسی به API استفاده می کند.

## امکانات گسترده

DRF از امکانات متنوعی برای ایجاد وب سرویس های پیچیده پشتیبانی می کند. این شامل امکاناتی مانند فیلترها، جستجوها، مرتب سازی و صفحه بندی اطلاعات می شود که به برنامه نویسان امکان می دهد تا به راحتی به مدیریت داده ها پرداخته و تنظیمات مورد نیاز را اعمال کنند البته برای استفاده بهتر از drf باید از هاست جنگو مناسب استفاده کنید تا سرعت و عملکرد مناسبی داشته باشید.

با توجه به این ویژگی ها و امکانات، Django Rest Framework یکی از فریمورک های برتر در زمینه توسعه وب سرویس های RESTful است که برنامه نویسان را در فرایند ساخت و مدیریت وب سرویس ها یاری می دهد. این فریمورک به برنامه نویسان این امکان را می دهد تا با سادگی و امنیت بالا وب سرویس هایی ایجاد کنند که به سرعت و با کیفیت بالا بتوانند با نیازهای کاربران سازگاری یابند.

## نتیجه گیری

به عنوان یک فریمورک قدرتمند و انعطاف پذیر، Django Rest Framework امکانات بی نظیری برای توسعه وب سرویس های RESTful ارائه می دهد. از سهولت در ایجاد و مدیریت وب سرویس ها گرفته تا امکانات امنیتی و مستندسازی، این فریمورک باعث می شود که توسعه دهندگان بتوانند به راحتی وب سرویس های پیچیده و کارآمدی بسازند.



## فصل دوم: ابزارها و روش‌های مورد استفاده

### پایگاه داده SQLite

SQLite یک سیستم مدیریت پایگاه داده رابطه‌ای (RDBMS) کوچک، سبک، قابل حمل و متن باز است که به طور گسترده در برنامه‌های کاربردی و وب سایت‌ها استفاده می‌شود. برخلاف پایگاه‌های داده رابطه‌ای سنتی که معمولاً در یک سرور مجزا اجرا می‌شوند، SQLite تمام داده‌ها و تابع‌های پایگاه داده را در یک فایل واحد ذخیره می‌کند. این ویژگی آن را بسیار سبک، قابل حمل و مناسب برای استفاده در برنامه‌های کاربردی کوچک و متوسط می‌کند. در پروژه مورد استفاده برای کاهش پیچیدگی‌های پروژه از SQLite استفاده شده است، همچنین جنگو قابلیت سینک شدن با پایگاه داده postgresql را دارد.

در ادامه به توضیح جزئیات SQLite می‌پردازیم:

۱. **ساختار پایگاه داده:** SQLite از مدل رابطه‌ای پیروی می‌کند و داده‌ها را در جداول متشکل از سطرها و ستون‌ها ذخیره می‌کند. هر جدول می‌تواند شامل انواع داده‌های مختلف از جمله عددی، رشته‌ای، تاریخ و زمان، و غیره باشد. SQLite از کلیدهای اصلی (Primary Key) و کلیدهای خارجی (Foreign Key) برای برقراری ارتباط بین جداول پشتیبانی می‌کند.

۲. **زبان پرس و جو:** SQLite از زبان استاندارد SQL (Structured Query Language) برای ایجاد، خواندن، به‌روزرسانی و حذف داده‌ها (CRUD) استفاده می‌کند. این شامل دستورات CREATE TABLE، INSERT، SELECT، UPDATE و DELETE است. SQLite همچنین از دستورات پیچیده‌تر SQL مانند JOIN، GROUP BY، HAVING و دیگر موارد پشتیبانی می‌کند.

۳. **تراکنش‌های اتمیک:** SQLite از تراکنش‌های اتمیک (Atomic Transactions) پشتیبانی می‌کند که به معنای آن است که یک سری از عملیات پایگاه داده باید به طور کامل موفقیت‌آمیز باشند یا در غیر این صورت، هیچ تغییری در داده‌ها اعمال نشود. این ویژگی یکپارچگی داده‌ها را تضمین می‌کند.

۴. **قفل‌گذاری خودکار:** SQLite از قفل‌گذاری خودکار (Automatic Locking) برای جلوگیری از تداخل داده‌ها در برنامه‌های چندنخی استفاده می‌کند. این امر اطمینان می‌دهد که تنها یک کاربر یا نخ در هر زمان می‌تواند داده‌ها را تغییر دهد.

۵. **عدم وابستگی به سرور:** SQLite یک پایگاه داده بدون سرور است و به جای اجرا در یک سرور مجزا، در فضای حافظه برنامه کاربردی بارگذاری می‌شود. این ویژگی آن را بسیار سبک و قابل حمل می‌کند.

۶. **قابلیت حمل:** پایگاه داده SQLite در یک فایل واحد ذخیره می‌شود که می‌تواند به راحتی بین سیستم‌های مختلف منتقل شود. این ویژگی آن را ایده‌آل برای برنامه‌های کاربردی سیار و وب سایت‌هایی می‌کند که نیاز به ذخیره‌سازی داده‌های کاربر در سمت کلاینت دارند.

۷. **عدم نیاز به پیکربندی:** SQLite نیاز به پیکربندی پیچیده ندارد و به محض نصب، آماده استفاده است. این ویژگی آن را برای توسعه‌دهندگان و برنامه‌نویسان جذاب می‌کند.

۸. **کراسپلتفرم:** SQLite در بسیاری از سیستم‌عامل‌ها و پلتفرم‌ها از جمله لینوکس، ویندوز، macOS، Android و iOS قابل استفاده است.

۹. **رایگان و متن باز:** SQLite یک پروژه متن باز و رایگان است که توسط جامعه بزرگی از توسعه‌دهندگان حمایت می‌شود.

۱۰. **امنیت:** SQLite از رمزنگاری پشتیبانی نمی‌کند، اما می‌توان با استفاده از ابزارهای خارجی، داده‌های ذخیره شده در آن را رمزگذاری کرد.

SQLite به دلیل سادگی، سرعت، قابلیت حمل و عدم نیاز به پیکربندی پیچیده، در بسیاری از برنامه‌های کاربردی دسکتاپ، موبایل، وب و حتی در برخی سیستم‌های بزرگ استفاده می‌شود. با این حال، برای پروژه‌های بزرگ‌تر و پرتراکنش‌تر، ممکن است نیاز به استفاده از پایگاه‌های داده قدرتمندتری مانند MySQL، PostgreSQL یا Oracle باشد.

## celery

سری (Celery) یک سیستم عامل توزیع شده برای پردازش وظایف آسنکرون (Asynchronous Task Queue) است که در برنامه‌های متن‌باز پایتون استفاده می‌شود. سری به برنامه‌نویسان این امکان را می‌دهد تا وظایف زمان‌بر یا زمینه‌بر را به صورت آسنکرون و موازی پردازش کنند، بدون آنکه برنامه اصلی را مسدود کنند. در پروژه مورد بحث از سِری برای فرایند ارسال ایمیل استفاده شده است.

### ویژگی‌های اصلی سری عبارتند از:

۱. **توزیع شده:** سری یک سیستم توزیع شده است که می‌تواند وظایف را بین چندین ماشین یا نود پردازش کند. این ویژگی امکان مقیاس‌پذیری و افزایش توان پردازش را فراهم می‌کند.

۲. **صف وظایف:** سری از یک برنامه صف وظایف (Task Queue) استفاده می‌کند که در آن وظایف آسنکرون برای پردازش در صف قرار می‌گیرند. سری از چندین سیستم صف مختلف مانند Amazon SQS, Redis, RabbitMQ و ... پشتیبانی می‌کند.

۳. **اجرای وظایف پیچیده:** سری می‌تواند وظایف پیچیده‌ای را اجرا کند، از جمله وظایف تکراری (Periodic Tasks)، وظایف زنجیره‌ای (Chained Tasks)، وظایف با روابط پیچیده و وظایفی که نیاز به پیگیری وضعیت دارند.

۴. **تعریف صریح وظایف:** در سری، وظایف به صورت صریح در کدهای پایتون تعریف می‌شوند. این امر خوانایی، نگهداری و آزمون‌پذیری کد را افزایش می‌دهد.

۵. **مانیتورینگ و مدیریت:** سری ابزارهای مانیتورینگ و مدیریت قدرتمندی را ارائه می‌دهد که به شما امکان می‌دهد وضعیت وظایف، صف‌ها و نودهای اجرایی را بررسی کنید و آن‌ها را مدیریت کنید.

۶. **انعطاف‌پذیری:** سری انعطاف‌پذیر است و می‌تواند با برنامه‌های پایتون مختلف و چارچوب‌های وب مانند Flask, Django, Pyramid و ... ادغام شود.

۷. **خطاهای وظایف:** سری مکانیزم‌های مدیریت خطا را ارائه می‌دهد که به شما امکان می‌دهد خطاهای وظایف را ردیابی، ذخیره و پردازش کنید.

۸. **تحویل مطمئن پیام‌ها:** سری از تحویل مطمئن پیام‌ها (Reliable Message Delivery) پشتیبانی می‌کند، که اطمینان می‌دهد هیچ وظیفه‌ای از دست نمی‌رود یا دوبار پردازش نمی‌شود.

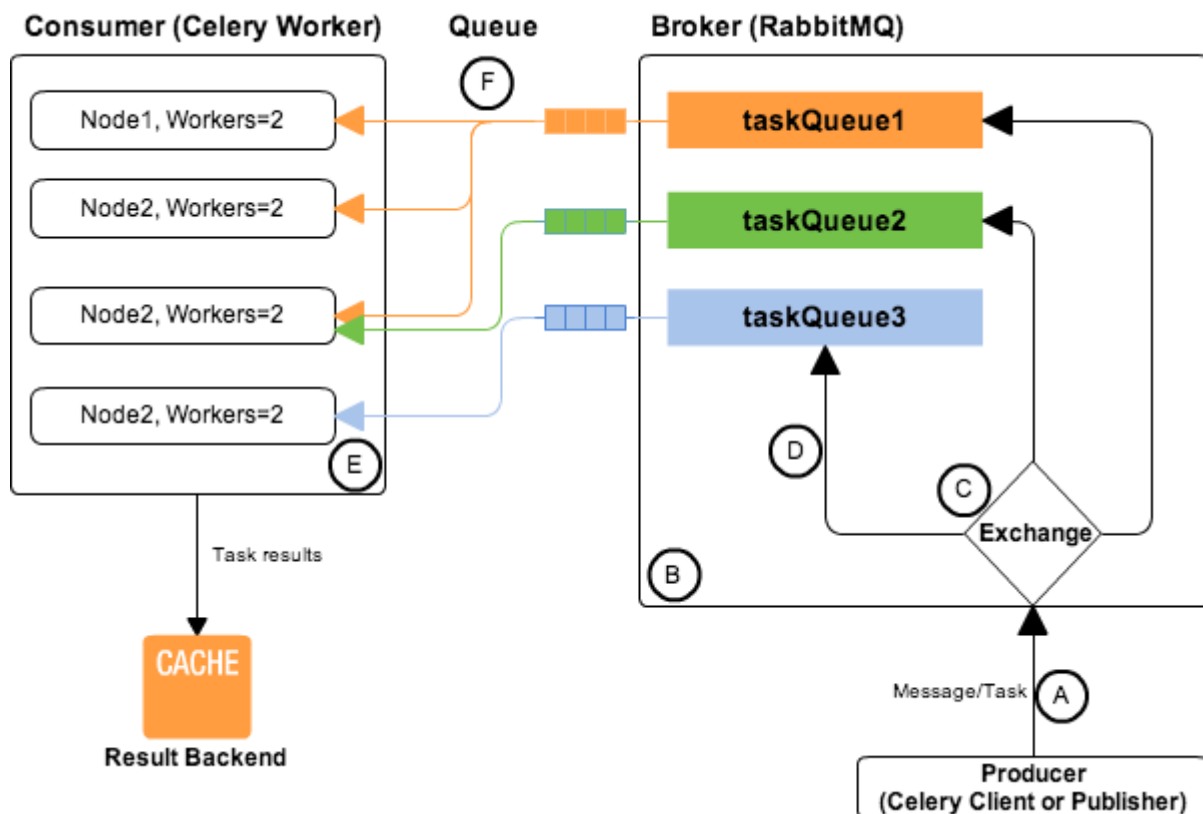
سری معمولاً در برنامه‌های وب، سیستم‌های پردازش داده، سیستم‌های فایل، ارسال ایمیل و هر جایی که نیاز به پردازش آسنکرون و موازی وظایف وجود دارد، استفاده می‌شود. با استفاده از سری، می‌توانید برنامه‌های پاسخگوتر، کارآمدتر و مقیاس‌پذیرتری داشته باشید.

در مواقعی که ترافیک روی سرور زیاد باشد، برای کارهایی که نیاز به پاسخ دهی آنی ندارند، مانند پردازش تصویر یا خروجی پرس و جوهای پیچیده database، پاسخ دهی به کلاینت دچار وقفه خواهد شد. یکی از راه حل‌های این مشکل استفاده از task queue هایی مانند celery است. celery با استفاده از یک message broker نظیر redis یا RabbitMQ (از جمله message broker های شناخته شده برای صف بندی پیام یا message queueing)، task ها را در صف (ها) قرار می‌دهد (queueing) و به محض اینکه سرور قادر به پاسخگویی بود درخواست ها را به سمت سرور هدایت می‌کند.

سلری (Celery) با زبان python و بصورت open source نوشته شده است و زبان‌های برنامه نویسی مانند php، Ruby و java نیز می‌توانند از آن بهره‌مند شوند.

سلری یک task queue غیر همروند (asynchronous) است که بر اساس بهترین راه حل‌های مسائل سازمان‌ها ایجاد شده است و پروتکل AMQP را پیاده سازی می‌کند. این پروتکل در مواقعی که سرعت تولید کننده (producer) و سرعت مصرف کننده (consumer) پیام یکسان نباشد، به منظور نگهداری پیام‌ها توسط یک یا چند صف (queue) کاربرد دارد. یکی از مهمترین مشکلات پروتکل AMQP پیچیدگی زیاد در پیکربندی و نگهداری آن در پروژ است. از آنجا که مراحل نصب celery، پیکربندی و استفاده از آن آسان است محبوبیت زیادی را در بین توسعه دهندگان کسب کرده است.

در حقیقت می‌توان celery را به عنوان تولیدکننده و مصرف کننده message های message broker در نظر گرفت. در celery کلاینت را producer و worker را consumer می‌نامند. یک task یا message متشکل از header و body است که message payload یا body شامل دیتایی است که به سمت مصرف کننده هدایت می‌شود.



در انتها به معرفی اجزای Celery و شرح وظیفه هرکدام می پردازیم:

- **تولیدکننده پیام (Producer):** یک کاربر اپلیکیشنی است که پیامها را ارسال می کند.
- **واسطه (Broker):** پیام ها را از Producer دریافت می کند و آنها را به سمت Consumer هدایت می کند. Broker می تواند شامل یک یا چند صف باشد.
- **تبادل (Exchange):** یک Producer پیامها را از طریق تبادل (Exchange) ارسال می کند. Exchange ها پیام ها را دریافت می کنند و آنها را در صف ها قرار می دهند.
- **صف (Queue):** یک بافر است که پیامها در آن دخیره می شوند.
- **قیدها (Bindings):** قوانین هستند که exchange ها برای مسیریابی از آنها استفاده می کنند.
- **کلیدهای مسیریابی (Routing Keys):** اتصال ممکن است یک Routing Keys اختیاری داشته باشد که exchange از آن برای انتقال پیام به صف استفاده کند.
- **مصرف کننده پیام (Consumer):** یک Consumer اپلیکیشنی است که پیام ها را دریافت کرده و آنها را پردازش می کند.

## احراز هویت JWT

و روش کلی و پرکاربرد اعتبارسنجی سمت سرور، برای برنامه‌های سمت کاربر وب وجود دارند:

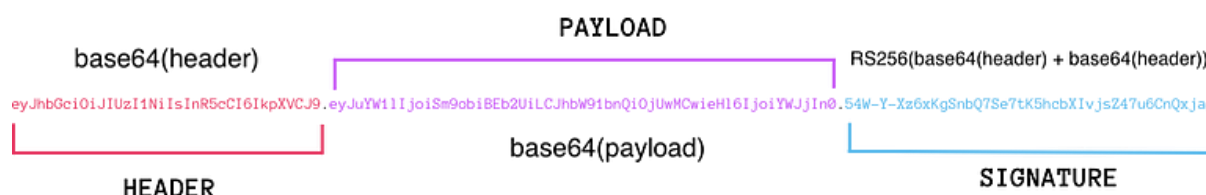
روش اول Cookie-Based Authentication که پرکاربردترین روش بوده و در این حالت به ازای هر درخواست، یک کوکی جهت اعتبارسنجی کاربر به سمت سرور ارسال می‌شود (و برعکس).

روش دوم Token-Based Authentication که بر مبنای ارسال یک توکن امضا شده به سرور، به ازای هر درخواست است.

### jwt

یا JSON Web Token یک استاندارد باز (RFC 7519) برای ایجاد توکن‌های دسترسی خودگواه (Self-contained) است. این توکن‌ها برای اهداف مختلفی مانند احراز هویت و انتقال داده‌های امن بین طرفین استفاده می‌شوند. JWT مزایای زیادی نسبت به روش‌های سنتی مدیریت احراز هویت مانند استفاده از جلسات (Session) دارد.

ساختار JWT شامل سه بخش اصلی است که با نقطه (.) از هم جدا می‌شوند:



۱. Header: این بخش شامل دو قسمت الگوریتم رمزنگاری مورد استفاده (مانند HMAC SHA256 یا RSA) و نوع توکن (JWT) است.

۲. Payload: این بخش شامل ادعاهای (Claims) مورد نظر است. ادعاها داده‌هایی هستند که توسط صادرکننده توکن تعیین می‌شوند، مانند شناسه کاربر، نقش‌های دسترسی، زمان انقضا و موارد دیگر. ادعاها دو نوع هستند: ادعاهای رسمی (Reserved Claims) که توسط استاندارد JWT تعریف شده‌اند و ادعاهای سفارشی که توسط صادرکننده توکن تعریف می‌شوند.

۳. Signature: این بخش از کدگذاری بخش‌های Header و Payload با استفاده از کلید رمزنگاری و الگوریتم مشخص شده در Header بدست می‌آید. این امضا برای تضمین امنیت و تمامیت داده‌های JWT استفاده می‌شود.

آخرین و مهم‌ترین بخش JWT ها Signature است، در حقیقت Signature تایید می‌کند که آیا داده‌ای که از سمت کاربر برگشته است بهم ریخته و یا دستکاری شده است؟

الگوریتمی که ما برای رمزنگاری در Header داده‌ایم را می‌گیرد، header و Payload را ترکیب می‌کند و آن‌ها را دوباره رمزنگاری می‌کند و بار دیگر با کلیدی که به Signature داده ایم رمزنگاری می‌کند و اگر کسی مقدار رمزگذاری شده را تغییر دهد خطایی دریافت می‌کنیم که به ما نشان می‌دهد Signature معتبر نمی‌باشد.

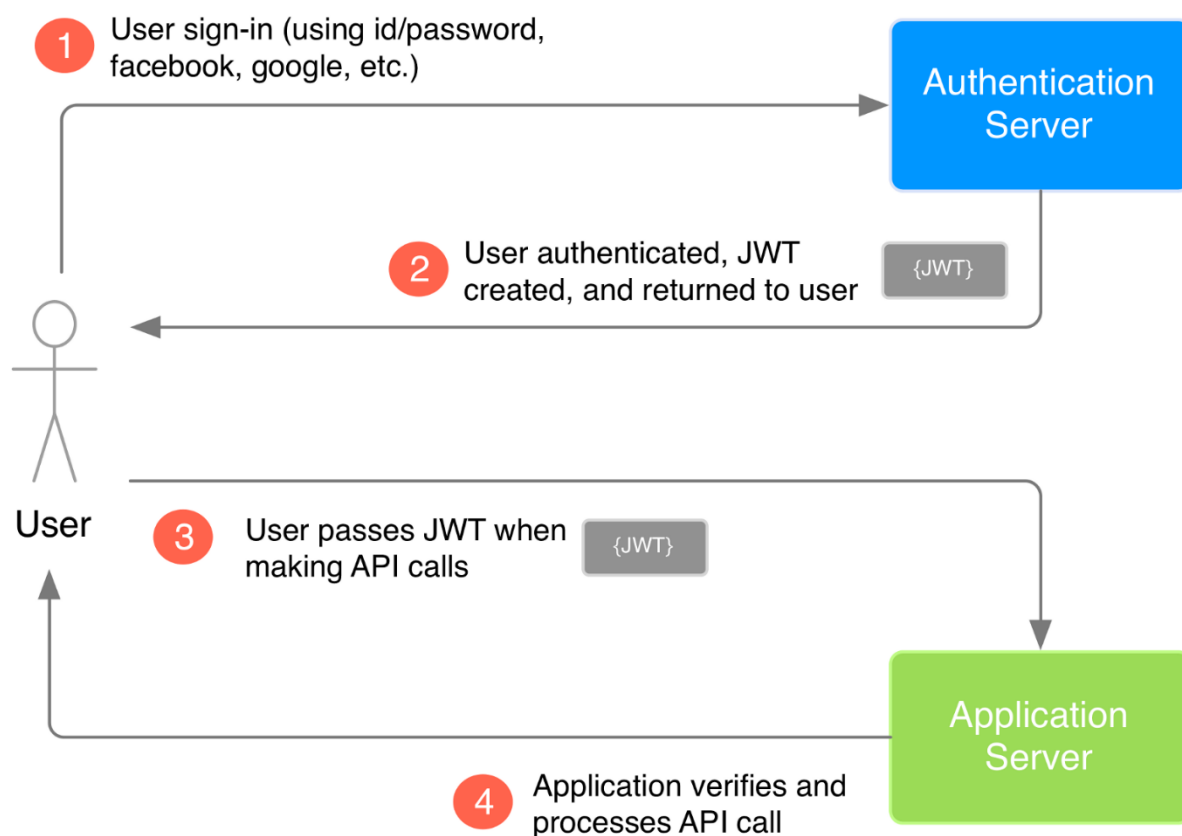
Signatureها نیز ساختاری شبیه به زیر دارند.

```

HMACSHA256 (
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload),
    secret)

```

فرآیند کار JWT به این صورت است:



۱. کلاینت (مرورگر وب یا برنامه) درخواستی را برای احراز هویت به سرور ارسال می‌کند.

۲. سرور با بررسی اعتبار اطلاعات احراز هویت، یک JWT جدید صادر می‌کند که شامل ادعاهای لازم (نام کاربری، نقش‌ها و ...) است.

۳. این JWT به کلاینت ارسال می‌شود.

۴. از این پس، کلاینت JWT را در هدرهای درخواست‌های بعدی قرار می‌دهد تا هویت خود را برای سرور اثبات کند.
۵. سرور JWT را دریافت و امضای آن را با استفاده از کلید رمزنگاری بررسی می‌کند. در صورت اعتبار، درخواست پردازش می‌شود.

### مزایای استفاده از JWT عبارتند از:

۱. توکن‌های خودگواه: JWT حاوی تمام اطلاعات لازم برای احراز هویت است و نیازی به ذخیره اطلاعات جلسه در سرور نیست.
۲. امنیت بالا: با استفاده از امضای دیجیتال، هرگونه تغییر در JWT قابل ردیابی است.
۳. انعطاف‌پذیری: JWT می‌تواند ادعاهای سفارشی را حمل کند و در معماری‌های مختلف مانند وب، موبایل یا میکروسرویس‌ها استفاده شود.
۴. مقیاس‌پذیری: از آنجایی که اطلاعات جلسه در سرور ذخیره نمی‌شود، سرورها می‌توانند بدون نگرانی از از دست رفتن اطلاعات جلسه، مقیاس‌پذیری را افزایش دهند.

با این حال، JWT هم مانند هر فناوری دیگری دارای برخی محدودیت‌ها و نکات امنیتی است که باید در نظر گرفته شود، مانند مدیریت کلیدهای رمزنگاری، محدودیت اندازه توکن و نحوه رد اعتبار توکن‌ها. در مجموع، JWT یک راه حل ساده و کارآمد برای احراز هویت و انتقال داده‌های امن در برنامه‌های مدرن است که به طور گسترده در معماری‌های وب، موبایل و میکروسرویس‌ها استفاده می‌شود.



## signals

در جنگو (Django)، سیگنال‌ها (Signals) یک مکانیزم ارتباطی هستند که به شما امکان می‌دهند به رویدادهای مختلف در اپلیکیشن واکنش نشان دهید و کدهای خاصی را در پاسخ به این رویدادها اجرا کنید. سیگنال‌ها به نوعی مشابه Event-Driven Programming هستند و یک راه کارآمد برای جداسازی مسئولیت‌ها (Separation of Concerns) و افزایش انعطاف‌پذیری کد محسوب می‌شوند.

جنگو دارای سیگنال‌های از پیش تعریف شده‌ای است که به رویدادهای مختلف مانند ذخیره، حذف یا آپدیت مدل‌ها، ارسال درخواست‌های HTTP، شروع و پایان پردازش درخواست‌ها و غیره مرتبط هستند. علاوه بر این، شما می‌توانید سیگنال‌های سفارشی خود را نیز تعریف کنید.

فرآیند کار با سیگنال‌ها در جنگو به این شرح است:

۱. **تعریف سیگنال:** ابتدا باید یک سیگنال را در کدتان تعریف کنید. این کار را می‌توانید با استفاده از کلاس `Signal` از ماژول `django.dispatch` انجام دهید.

۲. **ارسال سیگنال:** در نقطه‌ای از کد که می‌خواهید رویداد مربوطه رخ دهد، سیگنال را ارسال (`emit`) می‌کنید. برای این کار از متد `send` سیگنال استفاده می‌شود.

۳. **گیرنده سیگنال (Signal Receiver):** تابع یا متدی که باید در پاسخ به سیگنال اجرا شود، گیرنده سیگنال نامیده می‌شود. این تابع یا متد باید با استفاده از دکوراتور `@receiver` از ماژول `django.dispatch` پیکربندی شود تا بتواند سیگنال مورد نظر را دریافت کند.

۴. **اتصال گیرنده به سیگنال:** گیرنده‌های سیگنال باید به سیگنال مورد نظر متصل شوند تا بتوانند در پاسخ به آن اجرا شوند. این کار را می‌توان در `ready` method یک اپلیکیشن انجام داد.

### سیگنال‌ها در جنگو کاربردهای متنوعی دارند، از جمله:

- اعمال تغییرات بر روی داده‌ها یا مدل‌ها پس از رویدادهای خاص
- انجام عملیات جانبی مانند ارسال ایمیل، نوتیفیکیشن یا لاگ گرفتن
- اجرای کدهای مربوط به مانیتورینگ یا آمارگیری
- ارسال داده‌ها به سرویس‌های خارجی
- اعمال قوانین تجاری (Business Rules) یا محدودیت‌های اضافی بر روی داده‌ها

استفاده از سیگنال‌ها در جنگو باعث می‌شود کد شما تمیزتر، منعطف‌تر و قابل گسترش‌تر باشد، زیرا منطق اپلیکیشن از سایر بخش‌های آن مانند مدل‌ها، ویوها و فرم‌ها جدا می‌شود. همچنین، سیگنال‌ها امکان اضافه کردن رفتارهای جدید به اپلیکیشن بدون تغییر در کد اصلی را فراهم می‌کنند.



## فصل سوم: ابزارهای کنترل ورژن و تست

گیت (Git) و گیت‌هاب (GitHub) دو مفهوم مرتبط اما متفاوت در زمینه مدیریت کد منبع و همکاری در پروژه‌های نرم‌افزاری هستند.

### GIT

گیت یک سیستم کنترل نسخه توزیع‌شده است که برای ردیابی تغییرات در فایل‌های متنی در طول زمان استفاده می‌شود. گیت به توسعه‌دهندگان این امکان را می‌دهد که تغییرات در کد را به صورت محلی ردیابی کنند، شاخه‌های مختلف را ایجاد کنند، تغییرات را ادغام کنند و با سایر توسعه‌دهندگان همکاری کنند. گیت یک ابزار متن‌باز است که در سیستم‌عامل‌های مختلف قابل استفاده است.

#### ویژگی‌های اصلی گیت عبارتند از:

- ردیابی تغییرات در فایل‌های متنی
- مدیریت شاخه‌ها (Branching)
- ادغام تغییرات (Merging)
- کنترل نسخه توزیع‌شده
- عملیات‌های سریع و کارآمد

### GITHUB

گیت‌هاب یک سرویس میزبانی مبتنی بر ابر است که برای نگهداری مخازن گیت و همکاری در پروژه‌های نرم‌افزاری استفاده می‌شود. گیت‌هاب یک رابط کاربری وب ارائه می‌دهد که به توسعه‌دهندگان امکان می‌دهد تا مخازن گیت خود را میزبانی کنند، تغییرات را پیگیری کنند، مشکلات (Issues) را مدیریت کنند، درخواست‌های ادغام (Pull Requests) را بررسی کنند و با سایر توسعه‌دهندگان همکاری کنند.

#### ویژگی‌های اصلی گیت‌هاب عبارتند از:

- میزبانی مخازن گیت
- رابط کاربری وب برای مدیریت پروژه‌ها
- مدیریت مشکلات (Issues) و درخواست‌های ادغام (Pull Requests)
- همکاری و دسترسی چندنفره به مخازن

- ویکی و صفحات پروژه برای مستندسازی

- تکامل اجتماعی و اکوسیستم گسترده

در واقع، گیت یک ابزار محلی برای کنترل نسخه است، در حالی که گیت‌هاب یک سرویس میزبانی ابری است که از گیت استفاده می‌کند و امکانات بیشتری برای همکاری و مدیریت پروژه‌های نرم‌افزاری ارائه می‌دهد. توسعه‌دهندگان می‌توانند از گیت برای مدیریت کد محلی استفاده کنند و از گیت‌هاب برای میزبانی، همکاری و مدیریت پروژه‌های خود بهره ببرند.

لینک گیت‌هاب پروژه: [https://github.com/mahdipanahimpui/event\\_manager](https://github.com/mahdipanahimpui/event_manager)

## PostMan

Postman یک ابزار قدرتمند برای توسعه، تست و مستندسازی API های وب است. این ابزار رابط گرافیکی کاربرپسندی دارد که به توسعه‌دهندگان کمک می‌کند تا به راحتی با API ها کار کنند.

**کاربردهای اصلی Postman عبارتند از:**

۱. **ارسال درخواست‌های HTTP:** با Postman می‌توانید انواع درخواست‌های HTTP مانند GET, POST, PUT, DELETE و دیگر متدها را به سادگی ارسال کنید. این امکان به شما اجازه می‌دهد تا API های مختلف را آزمایش کنید.

۲. **تست API ها:** Postman ابزارهای قدرتمندی برای تست API ها ارائه می‌دهد. می‌توانید تست‌های اتوماتیک بنویسید، متغیرهای محیطی تعریف کنید، داده‌های ورودی را مدیریت کنید و نتایج را بررسی کنید.

۳. **مستندسازی API:** با Postman می‌توانید به راحتی مستندات API خود را تولید کنید. این ابزار امکان ایجاد مستندات با ساختار و قالب‌های استاندارد را فراهم می‌کند.

۴. **همکاری تیمی:** Postman امکانات همکاری تیمی را ارائه می‌دهد. شما می‌توانید مجموعه‌های درخواست‌ها (Collections) و محیط‌های (Environments) را با اعضای تیم به اشتراک بگذارید و روی یک پروژه به صورت همزمان کار کنید.

۵. **مدیریت محیط‌های توسعه:** با Postman می‌توانید محیط‌های مختلف توسعه را تعریف کنید، مانند محیط توسعه، آزمایشی و تولید. این امکان به شما کمک می‌کند تا API ها را در محیط‌های مختلف به راحتی آزمایش کنید.

۶. **اتوماسیون تست:** Postman با ارائه امکان نوشتن اسکریپت‌های تست با استفاده از JavaScript، به شما این امکان را می‌دهد تا فرآیند تست API ها را به طور کامل اتوماتیک کنید.

Postman در بسیاری از چرخه‌های توسعه API از جمله طراحی، ساخت، تست و مستندسازی استفاده می‌شود. این ابزار برای توسعه‌دهندگان، تستران و مهندسان API بسیار کاربردی است و می‌تواند باعث افزایش بهره‌وری و کیفیت API ها شود.

## فصل چهارم: ویژگی‌ها، امکانات و نکات پروژه

### ویژگی‌ها:

- استفاده از زبان پایتون و فریمورک جنگو و همچنین drf.
- قابلیت تولید، ذخیره و خروجی گرفتن از code QR برای مقاصدی مانند ثبت حضور در همایش و نشست‌ها و بررسی وعده غذایی.
- قابلیت export/import اطلاعات شرکت‌کنندگان به صورت فایل Excel.
- قابلیت تعریف رویدادها و نشست‌ها و قابلیت‌های ارسال ایمیل.
- قابلیت تعریف نظرسنجی‌های منعطف و ارسال لینک آن به صورت ایمیل و آمارگیری از اطلاعات آن.
- قابلیت ایجاد دو نوع نظرسنجی با پاسخ گزینه‌ای و یا متنی.
- قابلیت ایجاد ادمین‌ها و تغییر اطلاعات آن‌ها.
- قابلیت احراز هویت به روش jw token.
- قابلیت ارسال ایمیل از طریق mail serverهای جانبی..
- قابلیت اطلاع‌رسانی به صورت ایمیل به شرکت‌کنندگان.
- قابلیت جستجو و فیلترینگ اطلاعات در بخش‌های رویداد، نشست‌ها، شرکت‌کنندگان و ایمیل‌های ارسالی به وسیله query\_params.
- قابلیت ثبت‌نام فوری در سامانه توسط ادمین.
- ارسال ایمیل به صورت async به وسیله celery.
- استفاده از signals به منظور انجام برخی پروسه‌ها.
- قابلیت استفاده از پایگاه داده postgresql، که جز سریع‌ترین پایگاه داده‌ها می‌باشد.
- مدیریت تمامی مدل‌های پایگاه داده به وسیله پنل ادمین (Django admin panel)
- ویژگی‌های sorting, filtration, search در ادمین پنل.
- بهینه سازی جست‌وجوها در پایگاه داده برای اتصال کمتر به آن.
- استفاده از validationهای مختلف به منظور دریافت اطلاعات صحیح و جلوگیری از اختلال در روند پروسه‌ها.

## نکات پروژه:

- برای ارسال ایمیل از سرویس گوگل استفاده شده است که دارای محدودیت ارسال ۵۰۰ ایمیل در ۲۴ است، لذا نیاز به یک mail server جانبی می‌باشد، که در اکثر مواقع محتمل هزینه می‌باشد، که به راحتی می‌توان کانفیگ‌های آن را در فایل settings.py اعمال کرد.
  - پیشنهاد می‌شود تا پروژه در env ایزوله نصب شود و سپس موارد موجود در فایل requirements.txt نصب شود.
  - برای استفاده از پایگاه داده postgresql نیاز است تا کانفیگ‌های آن در فایل settings.py اعمال شود.
  - از آنجایی که برای ارسال ایمیل به صورت موازی از celery استفاده شده است، نیاز است تا پیش‌نیازهای آن یعنی rabbitmq بر روی سیستم لینوکسی نصب شود.
  - برای استفاده از celery پیشنهاد می‌شود تا از یک توزیع لینوکسی استفاده شود.
  - برای فعال کردن قابلیت ارسال ایمیل نیاز است تا celery به وسیله کامند زیر اجرا شود:
    - o `celery -A project worker --loglevel=info` و به جای project نام پروژه را قرار می‌دهیم.
  - اطلاعات موجود در qrcode به فرمت 'num{num}\_id{id}' می‌باشد.
  - وضعیت وعده غذایی (mail\_status)
    - o 0: رزروی موجود نیست
    - o 1: رزور موجود است
    - o 2: وعده غذایی دریافت شده است.
  - موارد استفاده از qrcode تنها برای دریافت شناسه شرکت‌کننده می‌باشد و کاربرهای آن از قبیل ثبت حضور و بررسی وعده غذایی بر عهده مصرف‌کننده api می‌باشد، تا تغییرات را اعمال کند.
  - استفاده از api ها نیازمند مجوزهای خاصی می‌باشد که در توضیحات url ها یادآوری شده است، انواع مجوزها:
    - o IsAuthenticated
    - o IsSuperUser
    - o AllowAny or []
  - برای ایمپورت کردن اطلاعات شرکت‌کنندگان به صورت فایل xlsx باید نام فیلدها در ستون اعمال شود.
- Method: POST(permission\_classes = [IsSuperUser])**
- URL: /events/<int:event\_id>/import/**
- در صورتی که حتی اطلاعات یک سطر مشکل داشته باشد، هیچ کدام از اطلاعات فایل وارد نخواهد شد.

- مقادیر

"ACCESS\_TOKEN\_LIFETIME": timedelta(days=3) o

"REFRESH\_TOKEN\_LIFETIME": timedelta(days=7) o

در فایل settings.py قابل تغییر می‌باشد.

در صورت دریافت خطای cors به لینک زیر مراجعه کنید.

<https://stackoverflow.com/questions/35760943/how-can-i-enable-cors-on-django-rest-framework>

## انواع ارسال ایمیل:

- ارسال ایمیل به شرکت کنندگان مختلف به وسیله url زیر و فیلتر کردن آدرس های ایمیل به وسیله query parms  
**Method: POST (permission\_classes = [IsAuthenticated])**  
**URL: /events/<int:event\_id>/send\_email/**
- ارسال ایمیل ثبت حضور در کنفرانس برای شرکت کنندگان، در صورتی که مقدار attendance\_time شرکت کنندگان مقداری باشد، (البته نیاز است مقدار sending\_attendace\_email مدل event به حالت true باشد).
- ارسال ایمیل ثبت حضور در کنفرانس برای شرکت کنندگان، در صورتی که مقدار attendance\_time شرکت کنندگان مقداری باشد، (البته نیاز است مقدار sending\_attendace\_email مدل event به حالت true باشد).
- ارسال ایمیل ثبت حضور در نشست برای شرکت کنندگان، در صورتی که شرکت کننده به وسیله فیلد add\_participant به نشست اضافه شود، (البته نیاز است مقدار sending\_attendace\_email مدل meeting به حالت true باشد).

- ارسال ایمیل حاوی لینک نظرسنجی رویداد به وسیله url زیر:

**Method: GET (permission\_classes = [IsSuperUser])**

**URL: /events/<int:event\_id>/send\_surveys/**

- ارسال ایمیل حاوی لینک نظرسنجی رویداد به وسیله url زیر:

**Method: GET (permission\_classes = [IsSuperUser])**

**URL: /events/<int:event\_id>/meetings/<int:meeting\_id>/send\_surveys/**

- پس از ارسال ایمیل نظرسنجی مقدار suvey\_email\_sent در مدل event و meeting به صورت خودکار true خواهد شد.

- نکته: لینک نظرسنجی موجود در ایمیل باید با مصرف کننده aip به توافق برسد. برای تغییر لینک نظرسنجی به آدرس های زیر مراجعه شود.

event\_manager\templates\email\_templates\meeting\_survey.html ○

event\_manager\templates\email\_templates\meeting\_survey.html ○

- برای مشاهده لاگ های ایمیل های ارسال شده:

**Method: GET (permission\_classes = [IsAuthenticated])**

**URL: /events/<int:event\_id>/email\_logs/**

**Method: GET (permission\_classes = [IsAuthenticated])**

**URL: /events/<int:event\_id>/email\_logs/<int:email\_log\_id>/**

## نحوه ایجاد انواع نظرسنجی:

دو نوع پاسخ برای نظرسنجی وجود دارد، ۱- پاسخ گزینه محور ۲- پاسخ متن محور  
۱- برای ایجاد متن نظرسنجی برای رویداد و هر نشست خاص از url های زیر استفاده می شود

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/surveys/

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/<int:meeting\_id>/surveys/

۲- برای ایجاد گزینه برای survey ها در صورتی که نظرسنجی پاسخ گزینه ای داشته باشد از url زیر استفاده می شود.

Method: POST (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/options/

۳- برای انتخاب گزینه توسط کاربر بدون هیچ گونه مجوزی از طریق url زیر:

Method: POST (permission\_classes = [AllowAny])

URL: /surveys/select\_options/

۴- در صورتی که نظرسنجی، گزینه ای نباشد و پاسخ آن به صورت متن باشد، نیازی به تعریف گزینه (option) نیست و برای افزودن پاسخ توسط شرکت کنندگان از طریق url زیر:

Method: POST

URL: /surveys/<int:survey\_id>/opinions/

۵- برای ارسال لینک نظرسنجی های ساخته شده به وسیله ایمیل:

Method: GET (permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/send\_surveys/

Method: GET (permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/meetings/<int:meeting\_id>/send\_surveys/

- پس از ارسال ایمیل نظرسنجی مقدار survey\_email\_sent در مدل event و meeting به صورت خودکار true خواهد شد.

- نکته: لینک نظرسنجی موجود در ایمیل باید با مصرف کننده aip به توافق برسد. برای تغییر لینک نظرسنجی به آدرس های زیر مراجعه شود.

event\_manager\templates\email\_templates\meeting\_survey.html ○

event\_manager\templates\email\_templates\meeting\_survey.html ○



۶- برای مشاهده آمار نظرسنجی‌های گزینه‌ای:

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/<option\_counter>/

۷- برای مشاهده پاسخ‌های نظرسنجی‌های دارای متن پاسخ:

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/<opinion\_counter>/



## فصل پنجم: api های url

### Admin panel url

به منظور استفاده از ادمین پنل جنگو از url زیر استفاده می شود.  
برخی پارامترها به هیچ وجه نباید تغییر و یا مقدار دهی شود که شامل:

(num) in participant model

(survey\_email\_sent) in event model

(survey\_email\_sent) in meeting model

Method: GET (in browser)

URL: /admin/

### Login urls

به دلیل آنکه سامانه درون سازمانی می باشد، ایجاد ادمین جدید تحت متن user تنها از طریق superuser صورت می گیرد و ادمین ها تنها اجازه لاگین کردن دارند، ادمین ها پس از افزوده شدن توسط superuser از طریق url های زیر می توانند لاگین کنند. پس از لاگین token دسترسی آن ها صادر و توسط فرانت اند در کوکی مرورگر ذخیره می شود، و بایستی قبل از فراخوانی برخی url ها در header جایگذاری شود.

Method: POST

URL: /login/

Multipart form body

Key	Required	Descriptions
username	*	
password	*	

Returns:

<refresh>

<access>

پس از انقضای access token می‌توان به وسیله url زیر access token را بروزرسانی کرد. پس از اتمام انقضای refresh token، نیاز است که کاربر دوباره از طریق /login/ لاگین کند.

Method: POST

URL: /login/token\_refresh/

Multipart form body

Key	Required	Descriptions
refresh	*	<refresh_token>

Returns:

<access>

## Profile urls

هر کاربری می‌تواند پروفایل خود را مشاهده کند که شامل موارد زیر می‌باشد،

Method: GET (permission\_classes = [IsAuthenticated])

URL: /profile/

Returns:

<id>  
 <is\_active>  
 <is\_staff>  
 <is\_superuser>  
 <username>  
 <password> (hashed)  
 <email>  
 <first\_name>  
 <last\_name>  
 <last\_login>  
 <date\_joined>  
 <groups>  
 <user\_permissions>

کاربر می‌تواند برخی اطلاعات خود را تغییر دهد، اگر که کاربر از طریق متد PUT اقدام به تغییر کند نیاز است فیلدهای \* از دوباره مقداردهی شود، در متد PATCH نیازی به مقدار دهی موارد الزامی نیست.

Method: PUT/PATCH (permission\_classes = [IsAuthenticated])

URL: /profile/

Multipart form body

Key	Required PUT	Descriptions
username	* unique	
password	*	
confirm_password	*	
email		
first_name		
last_name		

## Admins urls

از آنجایی که تنها superuser می تواند کاربر ایجاد کند، تنها superuser است که می تواند لیست userها را مشاهده کند. با استفاده از query param های جدول زیر می توان، لیست دلخواه را دریافت کرد.

مثال:

page=1&page\_size=2

order\_by=id

is\_active=false&ascending=false

search=mahdi

Method: GET (permission\_classes = [IsSuperUser])

URL: /admins/

Query_params:	Descriptions
page	
page_size	
order_by	Order by model fields: id, is_active, is_staff, is_superuser, username, email, first_name, last_name
ascending[=true]	Valid inputs: true, false (case-insensitive)
search	Search in fields: username, email, first_name, last_name
is_active	Valid inputs: true, false (case-insensitive)

Returns:

<count>

<next>

<previous>

<results> (list of objects)

<id>

<is\_active>

<is\_staff>

<is\_superuser>

<username>

<password> (hashed)

<email>

<first\_name>

<last\_name>

<last\_login>

<date\_joined>

<groups>

<user\_permissions>

---

تنها superuser می‌تواند ادمین جدید تحت متن user ایجاد کند. موارد \* الزامی می‌باشد.

Method: POST (permission\_classes = [IsSuperUser])

URL: /admins/

Multipart form body

Key	Required	Descriptions
username	* unique	
password	*	
confirm_password	*	
is_active[=true]		
email		
first_name		
last_name		

---

برای دریافت اطلاعات یک user

Method: GET (permission\_classes = [IsSuperUser])

URL: /admins/<int:id>/

Returns:

```

<id>
<is_active>
<is_staff>
<is_superuser>
<username>
<password> (hashed)
<email>
<first_name>
<last_name>
<last_login>
<date_joined>
<groups>
<user_permissions>

```

---

برای تغییر اطلاعات یک user

Method: PUT/PATCH (permission\_classes = [IsSuperUser])

URL: /admins/<int:id>/

Multipart form body

Key	Required PUT	Descriptions
username	* unique	
password	*	
confirm_password	*	
is_active[=true]		
email		
first_name		
last_name		

برای حذف اطلاعات یک user

Method: DELETE (permission\_classes = [IsSuperUser])

URL: admins/<int:pk>/

Hard delete



## Events urls

هر کاربر احراز هویت شده می تواند لیست رویدادهای ایجاد شده را مشاهده کند.  
همچنین از طریق برخی کوئری ها می تواند فیلترهایی اعمال کند.  
مثال:

order\_by=start\_date&is\_active=true  
search=cysp

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/

Query_params:	Descriptions
page	
page_size	
order_by	Order by model fields: id, name, is_active, end_date, start_date, created_at, note, sending_attendace_email, survey_email_sent
ascending[=true]	Valid inputs: true, false (case-insensitive)
search	Search fields: name, note
is_active	Valid inputs: true, false (case-insensitive)
sending_attendance_email	Valid inputs: true, false (case-insensitive)
survey_email_sent	Valid inputs: true, false (case-insensitive)

Returns:

```

<count>
<next>
<previous>
<results> (list of objects)
    <id>
    <name>
    <is_active>
    <start_date>
    <end_date>
    <created_at>
    <updated_at>
    <note>
    <sending_attendance_email>
    <survey_email_sent>

```

هر فرد احراز هویت شده‌ای می‌تواند رویداد ایجاد کند، موارد \* الزامی می‌باشد.

اگر مقدار sending\_attendace\_email=True باشد، زمانی که فیلد attendance\_time شرکت‌کننده‌ای (participant) در زمان حضور در روز رویداد مقداردهی شود، به صورت خودکار ایمیلی به شرکت‌کننده ارسال می‌شود و ثبت حضور وی را اطلاع رسانی می‌کند.

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/

Multipart form body

Key	Required	Descriptions
name	* unique	
start_date	*	
end_date	*	
sending_attendance_email	*	
note		

برای دریافت اطلاعات یک رویداد از url زیر استفاده می‌شود.

مقدار survey\_sent\_email نشان دهنده آن است که آیا بسته نظرسنجی مربوط به رویداد (و نه نشست‌ها) به تمامی شرکت کنندگان ارسال شده است یا خیر، در صورت ارسال نظرسنجی این مقدار به صورت خودکار true خواهد بود.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:id>/

Returns:

```

<id>
<name>
<is_active>
<start_date>
<end_date>
<created_at>
<updated_at>
<note>
<sending_attendance_email>
<survey_email_sent>

```

موارد زیر را می‌توان ویرایش کرد، در صورت استفاده از متد PUT در ویرایش، موارد \* الزامی می‌باشد.

Method: PUT/PATCH (permission\_classes = [IsAuthenticated])

URL: /events/<int:id>/

Multipart form body

Key	Required PUT	Descriptions
name	* unique	
start_date	*	
end_date	*	
sending_attendance_email	*	
note		

برای حذف رویداد صرفاً توسط superuser

Method: DELETE (permission\_classes = [IsSuperUser])

URL: /events/<int:pk>/

Hard delete

## Participants urls

برای دریافت اطلاعات تمامی شرکت‌کننده‌های ثبت شده در یک رویداد به وسیله آی‌دی رویداد از url زیر استفاده می‌شود. همچنین می‌توان با توجه به جدول زیر لیست را فیلتر کرد.  
مثال:

meal\_status=0&registered\_as=invidual&membership\_type=ut\_student

مقدار num به صورت خودکار مقداردهی می‌شود.

کیوآر کد هر شرکت‌کننده به صورت خودکار ساخته می‌شود و محتوی 'num{num}\_id{id}' می‌باشد.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/participants/

Query_params:	Descriptions
page	
page_size	
order_by	Order by: num, regestered_as, title, first_name, last_name, education_level, science_ranking, membership_type, mobile_phone_number, city, email_address, meal, attendance_time
ascending[=true]	Valid inputs: true, false (case-insensitive)
search	Search fields: first_name, last_name, mobile_phone_number, city, email_address
attendance_time	Valid inputs: none , not_none (case-sensitive)
meal_status	Valid inputs: 0,1,2 (0: not have, 1: has, 2: taken)
title	Valid inputs: mr, mrs, dr (case-sensitive)
regestered_as	Valid inputs: invidual, entity (case-sensitive)
education_level	Valid inputs: phd, md, phd_candinate, masters, masters_student, bachelors, bachelor_student, other (case-sensitive)
science_ranking	Valid inputs: professor, assistant_professor, associated_professor, other (case-sensitive)
membership_type	Valid inputs: ordinary_author, student_author, ut_student, ordinary_student, free_participant (case-sensitive)

Returns:

<count>

<next>

<previous>

<results> (list of objects)

<id>

<num>  
 <event>  
 <regestered\_as>  
 <title>  
 <first\_name>  
 <last\_name>  
 <education\_level>  
 <science\_ranking>  
 <mobile\_phone\_number>  
 <membership\_type>  
 <city>  
 <email\_address>  
 <meal>  
 <qr\_code>  
 <attendance\_time>  
 <created\_at>  
 <updated\_at>

هر کاربر احراز هویت شده‌ای می‌تواند به رویداد، شرکت کننده اضافه کند.  
ایمیل و شماره تماس شرکت کنندگان باید نسبت به هر رویداد متمایز باشد.

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/participants/

Multipart form body

Key	Required	Descriptions
regestered_as	*	Valid inputs: inividual, entity (case-sensitive)
title	*	Valid inputs: mr, mrs, dr (case-sensitive)
first_name	*	
last_name	*	
education_level	*	Valid inputs: phd, md, phd_candinate, masters, masters_student, bachelors, bachelor_student, other (case-sensitive)
science_ranking	*	Valid inputs: professor, assistant_professor, associated_professor, other (case-sensitive)
mobile_phone_number	*	phone number must be digits. (+) can be at first

membership_type	*	Valid inputs: ordinary_author, student_author, ut_student, ordinary_student, free_participant (case-sensitive)
city	*	
email_address	*	Unique with event
event_id	*	Unique with event
meal	*	Valid inputs: 0, 1, 2 (0: not have, 1:has, 2:gotten)

برای دریافت اطلاعات یک شرکت کننده در رویداد خاص از url زیر استفاده می شود.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/participants/<int:participant\_id>/

Returns:

```

<id>
<num>
<event>
<regerstered_as>
<title>
<first_name>
<last_name>
<education_level>
<science_ranking>
<mobile_phone_number>
<membership_type>
<city>
<email_address>
<meal>
<qr_code>
<attendance_time>
<created_at>
<updated_at>

```

برای تغییر اطلاعات یک شرکت کننده از url زیر استفاده می‌شود.  
 در صورت `sending_attendace_email=True` در `event` باشد، زمانی که فیلد `attendance_time` شرکت کننده‌ای (participant) در زمان حضور در روز رویداد مقداردهی شود، به صورت خودکار ایمیلی به شرکت کننده ارسال می‌شود و ثبت حضور وی را اطلاع رسانی می‌کند.

Method: PUT/PATCH (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/participants/<participant\_id>/

Multipart form body

Key	Required PUT	Descriptions
regestered_as	*	Valid inputs: inividual, entity (case-sensitive)
title	*	Valid inputs: mr, mrs, dr (case-sensitive)
first_name	*	
last_name	*	
education_level	*	Valid inputs: phd, md, phd_candinate, masters, masters_student, bachelors, bachelor_student, other (case-sensitive)
science_ranking	*	Valid inputs: professor, assistant_professor, associated_professor, other (case-sensitive)
mobile_phone_number	*	phone number must be digits. (+) can be at first
membership_type	*	Valid inputs: ordinary_author, student_author, ut_student, ordinary_student, free_participant (case-sensitive)
city	*	
email_address	*	Unique with event
event_id	*	Unique with event
attendance_time		a datetime
meal	*	Valid inputs: 0, 1, 2 (0: not have, 1:has, 2:gotten)

برای حذف شرکت کننده تنها توسط superuser

Method: DELETE (permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/participants/<participant\_id>/

Hard delete

## Import / Export file urls

خروجی اکسل از تمامی شرکت کنندگان یک رویداد

Method: GET(permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/export/

exports a xlsx file of participants

---

تنها superuser می تواند یک فایل xlsx شامل اطلاعات با فرمت زیر را وارد کند.

Method: POST(permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/import/

import a xlsx file to create participants, if one row fails to import, all rows fails(atomic creation)

max file size: 10MB

the xlsx file rows:

Key	Required	Descriptions
regestered_as	*	Valid inputs: invidual, entity (case-sensitive)
title	*	Valid inputs: mr, mrs, dr (case-sensitive)
first_name	*	
last_name	*	
education_level	*	Valid inputs: phd, md, phd_candinate, masters, masters_student, bachelors, bachelor_student, other (case-sensitive)
science_ranking	*	Valid inputs: professor, assistant_professor, associated_professor, other (case-sensitive)
mobile_phone_number	*	phone number must be digits. (+) can be at first
membership_type	*	Valid inputs: ordinary_author, student_author, ut_student, ordinary_student, free_participant (case-sensitive)
city	*	
email_address	*	Unique with event
meal	*	Valid inputs: 0, 1, 2 (0: not have, 1:has, 2:gotten)



## Meetings urls

برای دریافت لیست نشست‌های رویداد از url زیر استفاده می‌شود.

همچنین می‌توان نشست‌ها را بر اساس کوئری‌های زیر فیلتر کرد.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/

Query_params:	Descriptions
search	Search in fields: code, title, presenter, about_presenter, organizer, holding_place
survey_email_sent	Valid inputs: true, false (case-insensitive)
sending_attendace_email	Valid inputs: true, false (case-insensitive)

Returns:

<count>

<next>

<previous>

<results> (list of objects)

<id>

<event>

<code>

<title>

<holding\_time>

<presenter>

<about\_presenter>

<organizer>

<holding\_place>

<participants> (a list of participant ids)

<survey\_email\_sent>

<sending\_attendance\_email>

برای دریافت اطلاعات یک نشست خاص در رویداد مد نظر از url زیر استفاده می‌شود.  
 آی دی شرکت‌کنندگان (participant) در این نشست تحت یک آرایه نمایش داده می‌شود.  
 در صورتی که ایمیل نظرسنجی مربوط به این نشست (نه ایمیل نظرسنجی رویداد و یا سایر نشست‌ها) ارسال شود، مقدار survey\_email\_sent، true خواهد شد.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/<meeting\_id>/

Returns:

<id>  
 <event>  
 <code>  
 <title>  
 <holding\_time>  
 <presenter>  
 <about\_presenter>  
 <organizer>  
 <holding\_place>  
 <participants> (a list of participant ids)  
 <survey\_email\_sent>  
 <sending\_attendance\_email>

برای ایجاد نشست از url زیر استفاده می‌شود.  
 در صورتی که sending\_attendance\_email=true باشد، در صورت اضافه کردن شرکت‌کننده به نشست، ایمیل ثبت حضور در لحظه افزودن به نشست، حین برگزاری ارسال خواهد شد.

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/

Returns:

Key	Required	Descriptions
event	*	Valid inputs: event_id
code	*	
title	*	
holding_time	*	Datetime format
presenter	*	
holding_place	*	
sending_attendance_email	*	Default: false

برای افزودن شرکت کننده به نشست آی دی شرکت کنندگان (participant) تحت متن لیست و فیلد add\_participants انجام می شود.

شرکت کنندگان می توانند تکی تکی، ولی به صورت لیست اضافه شوند.

برای حذف شرکت کننده به نشست آی دی شرکت کنندگان (participant) تحت متن لیست و فیلد remove\_participants انجام می شود.

شرکت کنندگان می توانند تکی تکی، ولی به صورت لیست حذف شوند.

برای سایر تغییرات نیز می توان از url زیر استفاده کرد.

Method: PUT/PATCH (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/<meeting\_id>/

Returns:

Key	Required PUT	Descriptions
event	*	Valid inputs: event_id
code	*	
title	*	
holding_time	*	
presenter	*	
holding_place	*	
sending_attendance_email	*	Default: false
add_participants		Valid inputs: series of participants_id (ex: 1, 5, 48)
remove_participants		Valid inputs: series of participants_id (ex: 1, 5, 48)

برای حذف نشست از url زیر استفاده می شود.

Method: DELETE (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/<meeting\_id>/

hard delete

## Surveys urls

برای تمامی متن نظرسنجی‌های رویداد(فقط رویداد) از url زیر استفاده می‌شود.

در پاسخ این url مقدار meeting=null خواهد بود.

جواب نظرسنجی‌ها می‌تواند به صورت گزینه‌های مختلف و یا متن باشد، که در ادامه توضیح داده می‌شود.

**Method: GET**

**URL: /events/<int:event\_id>/surveys/**

Query_params:	Descriptions
page	
page_size	

**Returns:**

```

<count>
<next>
<previous>
<results> (list of objects)
    <id>
    <text>
    <event>
    <meeting>

```

---

برای دریافت متن نظرسنجی‌های مربوط به فقط نشست‌های رویدادها:

**Method: GET**

**URL: /events/<int:event\_id>/meetings/<int:meeting\_id>/surveys/**

Query_params:	Descriptions
page	
page_size	

**Returns:**

```

<count>
<next>
<previous>
<results> (list of objects)
    <id>
    <text>
    <event>
    <meeting>

```

برای ساخت متن نظرسنجی برای فقط برای رویداد

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/surveys/

Key	Required	Descriptions
text	*	
event	*	Valid inputs: event_id

برای ساخت متن نظرسنجی فقط برای نشست‌ها

Method: POST (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/meetings/<int:meeting\_id>/surveys/

Key	Required	Descriptions
text	*	
event	*	Valid inputs: event_id
meeting	*	Valid inputs: meeting_id

برای تغییر متن هر نظرسنجی چه متن نظرسنجی رویداد و چه نظرسنجی نشست‌های رویداد، از ای‌دی نظرسنجی استفاده می‌شود.

در صورت استفاده از متد PUT مقادیر \* الزامی می‌باشد.

Method: PUT/PATCH (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/

Key	Required PUT	Descriptions
text	*	
event	*	Valid inputs: event_id
meeting	* (required for meeting survey)	Valid inputs: meeting_id

برای دریافت اطلاعات متن هر نظرسنجی به وسیله آیدی آن نظرسنجی

Method: GET

URL: /surveys/<int:survey\_id>/

Returns

<id>  
<text>  
<event>  
<meeting>

---

برای حذف متن هر نظرسنجی

Method: DELETE (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/

hard delete

---

اگر نظرسنجی گزینه‌ای باشد، برای ساخت و اساین کردن گزینه‌ها url های زیر استفاده می‌شود.

برای مشاهده گزینه‌های هر نظرسنجی از url زیر استفاده می‌شود.

Method: GET

URL: /surveys/<int:survey\_id>/options/

Returns:

<count>  
<next>  
<previous>  
<results> (list of objects)  
    <id>  
    <option\_text>  
    <survey>

---

برای ساخت گزینه و اساین کردن گزینه به متن نظرسنجی مورد نظر از url زیر استفاده می‌شود.

Method: POST (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/options/

Key	Required	Descriptions
option_text	*	
survey	*	

برای مشاهده اطلاعات گزینه مورد نظر به وسیله آی دی گزینه، از url زیر استفاده می شود.

Method: GET (permission\_classes = [IsAuthenticated])

URL: surveys/options/<int:option\_id>/

Returns:

<id>  
<option\_text>  
<survey>

برای تغییر گزینه ها:

Method: PUT/PATCH (permission\_classes = [IsAuthenticated])

URL: /surveys/options/<int:option\_id>/

Key	Required PUT	Descriptions
option_text	*	
survey	*	

برای حذف گزینه ها:

Method: DELETE (permission\_classes = [IsAuthenticated])

URL: /surveys/options/<int:option\_id>/

hard delete

برای مشاهده تمامی اکشن های انتخاب گزینه توسط شرکت کنندگان از url زیر استفاده می شود.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/select\_options/

Query_params:	Descriptions
page	
page_size	

Returns:

<count>  
<next>  
<previous>  
<results> (list of objects)  
<id>

&lt;participants&gt;

&lt;survey&gt;

&lt;option&gt;

برای انتخاب گزینه نظرسنجی توسط شرکت کنندگان، بدون هیچ مجوزی.  
شرکت کنندگان از طریق لینک نظرسنجی ایمیل شده که حاوی آی دی رویداد و آی دی شرکت کننده می باشد، از این url استفاده می کند.

Method: POST (permission\_classes = [AllowAny])

URL: /surveys/select\_options/

Key	Required	Descriptions
participant	*	Valid inputs: participant id
survey	*	Valid inputs: survey id
Option	*	Valid inputs: option id

برای مشاهده اکشن انتخاب گزینه:

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/select\_options/&lt;int:select\_option\_id&gt;/

Returns:

&lt;id&gt;

&lt;participants&gt;

&lt;survey&gt;

&lt;option&gt;

تنها superuser اجازه تغییر و یا حذف گزینه های انتخاب شده را دارد.

Method: PUT/PATCH (permission\_classes = [IsSuperUser])

URL: /surveys/select\_options/&lt;int:select\_option\_id&gt;/

Key	Required	Descriptions
participant	*	Valid inputs: participant id
survey	*	Valid inputs: survey id
Option	*	Valid inputs: option id



تنها superuser اجازه تغییر و یا حذف گزینه‌های انتخاب شده را دارد.

Method: DELETE (permission\_classes = [IsSuperUser])

URL: /surveys/select\_options/<int:select\_option\_id>/

hard delete

می‌توان برای هر متن نظرسنجی به جای قرار دادن گزینه‌های انتخابی از پاسخ متنی استفاده کرد. برای مشاهده لیست پاسخ‌های اعمال شده از سمت شرکت‌کنندگان نسبت به یک متن نظرسنجی خاص از آی‌دی آن نظرسنجی استفاده می‌کنیم.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/opinions/

Returns:

```
<count>
<next>
<previous>
<results> (list of objects)
    <id>
    <opinion_text>
    <participant>
    <survey>
```

برای پاسخ به نظرسنجی به صورت متن از url زیر استفاده می‌شود. استفاده از url زیر برای همگان آزاد است. فیلد participant، که آی‌دی شرکت‌کننده می‌باشد، در لینک حاوی نظرسنجی به ایمیل شرکت‌کننده ارسال می‌شود.

Method: POST

URL: /surveys/<int:survey\_id>/opinions/

Key	Required	Descriptions
opinion_text	*	
participant	*	Valid inputs: participant id
survey	*	Valid inputs: survey id

برای مشاهده پاسخ متنی، نظرسنجی از url زیر استفاده می‌شود.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/opinions/<int:opinion\_id>/

Returns:

```
<id>
```

<option\_text>

<survey>

برای تغییر پاسخ متنی، نظرسنجی انجام شده تنها توسط superuser از url زیر استفاده می‌شود.

Method: PUT/PATCH (permission\_classes = [IsSuperUser])

URL: /surveys/opinions/<int:opinion\_id>/

Key	Required PUT	Descriptions
opinion_text	*	
participant	*	Valid inputs: participant id
survey		Valid inputs: survey id

برای حذف پاسخ متنی، نظرسنجی انجام شده تنها توسط superuser از url زیر استفاده می‌شود.

Method: DELETE (permission\_classes = [IsSuperUser])

URL: /surveys/opinions/<int:opinion\_id>/

hard delete

برای مشاهده آمار انتخاب یک نظرسنجی خاص که به پاسخ آن صورت گزینه‌ای است.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/<option\_counter>/

Returns:

<survey>

<results>

<option>

<option\_text>

<count>

برای مشاهده آمار انتخاب یک نظرسنجی خاص که به پاسخ آن صورت متنی است.

Method: GET (permission\_classes = [IsAuthenticated])

URL: /surveys/<int:survey\_id>/<opinion\_counter>/

Returns:

<survey>

<results>

<opinion\_count> (a list)

برای ارسال ایمیل حاوی لینک نظرسنجی فقط رویداد، توسط superuser.

Method: GET (permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/send\_surveys/

Returns:

<event>

<event\_name>

<sending\_to> (a list)

برای ارسال ایمیل حاوی لینک نظرسنجی فقط نشست مورد نظر، توسط superuser.

Method: GET (permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/meetings/<int:meeting\_id>/send\_surveys/

Returns:

<event>

<event\_name>

<sending\_to> (a list)

## Qrcode downloader url

برای دانلود تمامی کیوآرکدهای یک رویداد.

Method: GET (permission\_classes = [IsSuperUser])

URL: /events/<int:event\_id>/qrcodes/

Returns:

downloads the qrcodes

## Email logs urls

برای مشاهده لیست ایمیل‌های ارسال شده به کاربران از طرق مختلف  
قابلیت جست‌وجو بین ایمیل‌های ارسال شده به وسیله کوئری

مثال:

to=ali?

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/email\_logs/

Query_params:	Descriptions
page	
page_size	
search	Search fields: to, subject, body, title

Returns:

```
<count>
<next>
<previous>
<results> (list of objects)
    <id>
    <event>
    <meeting>
    <when>
    <to>
    <subject>
    <title>
```

---

برای مشاهده لاگ ایمیل

Method: GET (permission\_classes = [IsAuthenticated])

URL: /events/<int:event\_id>/email\_logs/<int:email\_log\_id>/

Returns:

```
<id>
<when>
<to>
<subject>
<title>
<event>
<meeting> (if exists)
```

## خلاصه عملکرد پروژه

سیستم مدیریت کنفرانس یک ابزار قدرتمند و کاربردی برای برگزارکنندگان رویدادها و کنفرانسها است. این سیستم با ارائه قابلیت‌های گسترده و کاربردی، فرآیند برنامه‌ریزی، سازماندهی و اجرای کنفرانسها را به طور قابل توجهی تسهیل می‌کند.

یکی از ویژگی‌های کلیدی این سیستم، امکان ایجاد و مدیریت رویدادها و نشست‌های مختلف است. برگزارکنندگان می‌توانند با استفاده از این قابلیت، جزئیات رویدادها و نشستها را به راحتی تعریف کنند و برنامه‌ریزی لازم را انجام دهند.

علاوه بر این، امکان ایجاد و مدیریت شرکت‌کنندگان در این سیستم فراهم شده است. برگزارکنندگان می‌توانند لیست شرکت‌کنندگان را تهیه کرده و اطلاعات مربوط به آنها را ثبت نمایند. این امر مدیریت و سازماندهی شرکت‌کنندگان را بسیار آسان می‌کند.

یکی دیگر از قابلیت‌های مهم این سیستم، امکان ایمپورت و اکسپورت اطلاعات است. این ویژگی به برگزارکنندگان اجازه می‌دهد تا داده‌های مربوط به رویدادها، نشستها و شرکت‌کنندگان را به راحتی وارد یا خارج کنند، که می‌تواند در مدیریت و تحلیل داده‌ها بسیار مفید باشد.

علاوه بر این، سیستم مدیریت کنفرانس امکان ایجاد انواع نظرسنجی‌ها را فراهم می‌کند. برگزارکنندگان می‌توانند نظرسنجی‌های مختلف را طراحی کرده و از شرکت‌کنندگان بازخورد دریافت نمایند. این ویژگی می‌تواند در ارزیابی و بهبود کیفیت رویدادها بسیار مفید باشد.

یکی دیگر از جنبه‌های مهم این سیستم، قابلیت ارسال ایمیل‌های اطلاع‌رسانی و نظرسنجی است. برگزارکنندگان می‌توانند از این ویژگی برای ارسال اطلاعیه‌ها، دعوت‌نامه‌ها و نظرسنجی‌ها به شرکت‌کنندگان استفاده کنند، که می‌تواند در افزایش میزان مشارکت و ارتباط با شرکت‌کنندگان بسیار مؤثر باشد.

علاوه بر این، سیستم مدیریت کنفرانس امکان ثبت حضور و غیاب شرکت‌کنندگان را فراهم می‌کند. این ویژگی می‌تواند در مدیریت و کنترل بهتر شرکت‌کنندگان و همچنین ارائه گزارش‌های دقیق‌تر بسیار مفید باشد.

در نهایت، استفاده از کیوآر کد در این سیستم، فرآیند ثبت‌نام و ورود شرکت‌کنندگان را بسیار آسان‌تر می‌کند. شرکت‌کنندگان می‌توانند با اسکن کردن کیوآر کد، به راحتی در رویدادها ثبت‌نام کرده و وارد شوند، که می‌تواند باعث صرفه‌جویی در زمان و افزایش کارایی فرآیند ثبت‌نام شود.

در مجموع، سیستم مدیریت کنفرانس با ارائه قابلیت‌های گسترده و کاربردی، یک ابزار قدرتمند برای برگزارکنندگان رویدادها و کنفرانسها محسوب می‌شود. این سیستم فرآیندهای مختلف را تسهیل کرده و به برگزارکنندگان امکان می‌دهد تا رویدادهای خود را به شکل بهتر و کارآمدتری برنامه‌ریزی، سازماندهی و اجرا کنند.



## منابع و مراجع

<https://docs.djangoproject.com/en/4.2/>

<https://www.django-rest-framework.org/topics/documenting-your-api/>

<https://docs.celeryq.dev/en/stable/>

<https://django-rest-framework-simplejwt.readthedocs.io/en/latest/>





پیوست‌ها

لینک گیت‌هاب پروژه

[https://github.com/mahdipanahimpui/event\\_manager](https://github.com/mahdipanahimpui/event_manager)



**University of Tehran  
College of Farabi  
Faculty of Engineering  
Department of Computer Engineering**

**Web API development for  
conference management**  
**(focusing on participant management)**

By:  
**Mahdi Panahi**

Under Supervision of:  
**Dr. Kazim Fouladi ghaleh**

**A Project Report as a Requirement for  
the Degree of Bachelor of Science in  
Computer Engineering**

**August 2024**





