# Zeitoon Server

سیستم به صورت کلی به دو بخش تقسیم میشود. سرور و سرویس ها. سرویس ها وظیفه ارائه خدمات مختلف را بر عهده دارند. سرور وظیفه مدیریت افزونه ها و انتقال اطلاعات بین آنها را بر عهده دارد. تنها راه ارتباطی سرویس ها با یکدیگر از طریق سرور است. تمامی اطلاعات با استفاده از پروتکل JSON منتقل میشوند تا توسط همه سرویس ها قابل فهم باشد. همچنین تمامی دستورات و اطلاعاتی که برای سرور ارسال میشوند هم در قالب JSON است.

## رویداد ها و دستورات

سرور دو روش برای انتقال اطلاعات را در اختیار سرویس ها قرار میدهد : Event و Command .

* Command : با استافاده از این قابلیت سرویس ها میتوانند یک command را که توسط سرویسی دیگر ارئه شده فراخوانی کنند، دیتا به آن بدهند و خروجی آن را دریافت کنند.
* Event : در این حالت سرویس درevent هایی که سرویس های دیگر ارائه کرده اند، مشترک میشود. سپس هنگامی که آن event (توسط ارائه دهنده آن) فراوانی شود، تمام اشتراک های ثبت شده مربوط به آن فراخوانی میشوند.

Node ها : در سیستم موجودیتی مجازی به عنوان Node داریم که موجودیت فیزیکی ندارد. Node نماینده ای برای بخش های مختلف است که توسط سرویس ها مورد استفاده قرار میگیرند. به عبارت دیگر بخش هایی که میتوانند توسط دیگرات آدرس دهی شوند مانند Event ، command و config ها. نکته مهم در مورد Node ها روش نامگذاری آن ها است. اسم یک Node از دو بخش تشکیل میشود : group.field . که در این جا group یک اسم مجازی است که ربطی به سرویس ها و بخش های دیگر سیستم ندارد و صرفاً برای گروه بندی Node ها استفاده میشود. سرور یک گروه به نام \_core دارد که Node های خود را با آن دسته بندی میکند و سرویس ها مجاز به استفاده از آن نیستند.

command هایی که سرور برای مدیریت این بخش در اختیار سرویس ها قرار میدهد :

\_core.registerCommand ، \_core.registerEvent ، \_core.registerHook ، \_core.removeCommand ، \_core.removeEvent ، \_core.removeHook

## تنظیمات

قابلیت دیگری که سرور در اختیار سرویس ها قرار میدهد قابلیت ذخیره کردن تنظیمات است. هر سرویس میتواند تعداد نامحدودی داده در بخش تنظیمات و با اسم یکتا ایجاد و ذخیره کند. همچنین یک سرویس میتواند یک مقدار را به صورت public تعریف کند تا توسط بقیه سرویس ها قابل دسترسی باشد. همچنین در صورت نیاز میتواند آن را فقط خواندنی تعریف کند تا فقط خود آن سرویس بتواند مقدار آن را تغییر دهد.

command هایی که سرور برای مدیریت این بخش در اختیار سرویس ها قرار میدهد :

\_core.registerConfig ، \_core.removeConfig ، \_core.updateConfig ، \_core.configUpdated

## شبکه

سرور از طریق پروتکل TCP به سرویس ها متصل میشود به این صورت که سرور با اجرا پورت مشخص شده خود را باز میکند و منتظر اتصال سرویس ها به آن میماند. هنگامی که یک سرویس درخواست اتصال بدهد سرور به آن متصل میشود و درخواست معرفی را به آن سرویس ارسال میکند که سرویس باید ابتدا به آن جواب دهد تا بتواند بعدا با سرور کار کند.

همچنین برای این که سرویس ها بتوانند سرور را پیدا کنند ، سرور به یک پورت UDP هم متصل میشود تا اگر سرویس ها درخواست پیدا کردن سرور را در شبکه broadcast کردند سرور آن را دریافت کند و به آن جواب دهد تا سرویس ها از آدرس سرور مطلع شوند.

تمامی اطلاعات همان طور که گفته شد با JSON کد میشود. یک بسته داده دریافتی در واقع یک struct است که شامل چند فیلد ثابت متغیر است. یکی از فیلد های ثابت فیلد توع داده دریافتی است که چهار حالت دارد:

* Internal : برای داده هایی است که برای مدیریت اتصال و معرفی سرویس به سرور مورد استفاده قرار میگیرند
* Call : برای فراخوانی یک command
* Callback : برای برگرداندن مقدار بازگشتی یک command
* Fire : برای اجرای یک event

تمامی اطلاعات رد و بدل شده در این چهار دسته قرار میگیرند.