

GOGLAND + GO-SWAGGER

TDD style WEB API development

GOGLAND

- JetBrains公司的Go IDE，還在EAP開發階段：
<https://www.jetbrains.com/go/>
- 已具備Debug、Run Test、Refactoring功能，和VSCode加掛Go開發擴充套件不相上下。
- 據說正式版要收錢!!! >_<
<https://www.jetbrains.com/help/go/1.0/faq.html#d3e11>
- 如同該公司其他語言的IDE一樣，支援加裝其他語言的開發外掛：
<https://github.com/zalando/intellij-swagger>

GO-SWAGGER

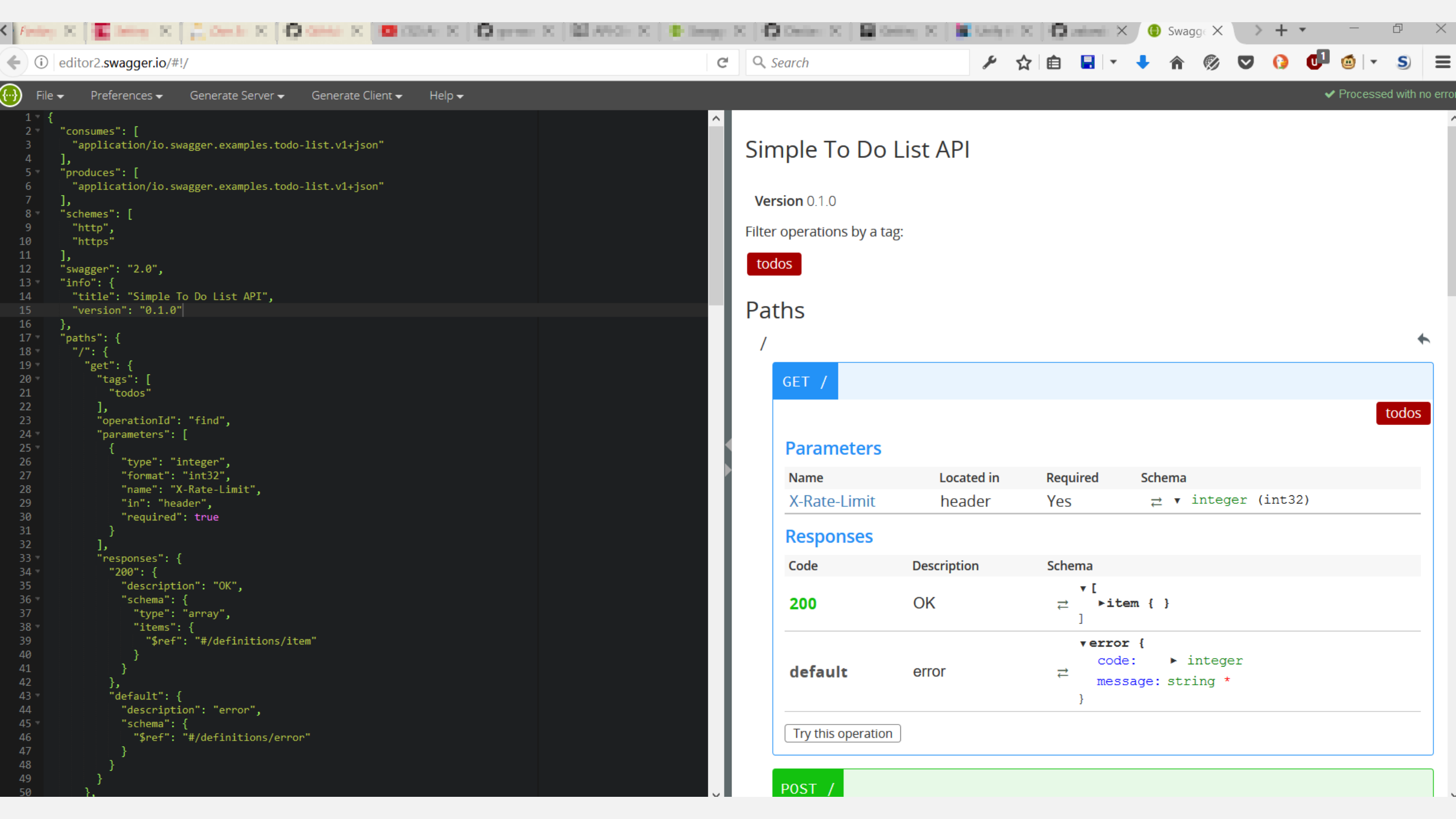
- Swagger ver2.0在Go語言上的實作框架，VMWare公司開源軟體：
<https://github.com/go-swagger/go-swagger>
- 可藉由API spec定義檔(swagger.yml)產生client和server的程式碼
- 產生的框架支援整合web api會用到的logging, authentication等middleware
<https://goswagger.io/use/middleware.html>
- 可藉由套用自訂範本來產生客製化的client/server實作程式碼
https://goswagger.io/use/template_layout.html

WHAT IS SWAGGER?

- **Swagger**：提供開發人員製作符合RESTful API公開規範(Open API specification, http://en.wikipedia.org/wiki/OpenAPI_Specification) 的工具
 1. 利用實際程式碼上的comment，產生Live Document(見下下頁)。
 2. 使用metadata定義檔來產生stub client/server的api實作。
- 使現在流行的前後端分離開發流程中，保有**Web API**規格一致性。
- **Design First** 的開發模式。

開發步驟

1. 設計好API的規範：swagger.yaml檔
<http://swagger.io/specification/>
2. 利用go-swagger的命令列工具產生server端程式碼並執行以便驗證server端建立正常。
3. 利用go-swagger的命令列工具對跑起來的server上所提供的Live Document metadata (swagger.json)，產生測試用的go client端。
4. 用建立起來的go client端寫測試，來逐步規範正確詳述的API spec
5. 修正Server端程式以便讓4.寫的測試通過。



DEMO

<http://github.com/windperson/goSwaggerDemo>

Q&A