vsCode Demo

Inhalt

[**1.** **Einführung** 2](#_Toc54334855)

[**2.** **Basis Connection** 3](#_Toc54334856)

[**3.** **AS3** 4](#_Toc54334857)

[**4.** **FAST Templates** 5](#_Toc54334858)

[**5.** **RPM Management** 6](#_Toc54334859)

[**6.** **OPEN API stuff …** 7](#_Toc54334860)

[**7.** **Convert // JSON <-> YAML & base64 de- encode** 9](#_Toc54334861)

[**8.** **Troubleshooting** 10](#_Toc54334862)

Start LTM-1 VM

Start LTM-2 VM

Client w VS Code

* F5 FAST extension
* YAML extension

1. **Einführung**

**Visual Studio Code** is a code editor redefined and optimized for building and debugging modern web and cloud applications.

GIT

<https://github.com/de1chk1nd/VScodeDemo>

VS Code Oberfläche

github Integration und Ordnerstruktur

<F1> – git: clone

extensions

<https://marketplace.visualstudio.com/items?itemName=DumpySquare.vscode-f5-fast>

**Alt:** Tool wurde weiterentwickelt. Früher war es für das Device Management (bigip hinzufügen, PWD Caching) gedacht, ATC Discovery, etc.

**Neu:**

* + ATC ***Management*** (Install RPM, pull from git)
    - DO
    - AS3 Call
    - FAST Templates
    - TS
  + YAML <> JSON Converter
  + base64 de- & encoding
  + (better) support from pushing from editor (right-click)
  + Open API “Support” (API Calls)

1. **Basis Connection**

* F5 extension im Tool zeigen
  + Information im unteren Teil (blau)
* Mit einer F5 Verbinden (LTM-1) [***menu links***; ***balken blau***]
  + <right-click> device *show device Info*
  + <right-click> device *login provider*

1. **AS3**

***Zeigen***

Verbundene F5 im unteren Bereich zeigen *blauer balken*

* Bigip
* Version

AS3 Menu links

* TASKS
* TENANT

<right-click> “*hostname*” im blauen Balken (Device Info – viele Wege führen nach Rom 😉 )

* Im json file <ctrl-a>, „*as3*“ eingeben, Template holen
  + ***SchemaFile*** ist eingebunden
    - ***Hover Over Content*** „Help“
    - ***Change Content*** Error-Message
    - ***Add Content*** Auto-Complete

***Anlegen & Testen***

Add Virtual Server : 10.1.10.150

Add Node : 212.227.165.10

* + CTRL-A > <right-click> > „POST AS3”
* Testen der Applikation (Browser: 10.1.10.150)
* Zeigen TMUI

***Verifizieren & löschen***

AS3 Menu

* AS3, Tenants, [Sample\_01]
  + Sample 01 App AS3 Call
  + <right-click>, expended AS3 Call mit default Werten
  + <right-click>, delete tenant löschen der App

1. **FAST Templates**

***Zeigen***

* Verbundene F5 im unteren Bereich zeigen (bigip, Version, etc)
* FAST Menu zeigen (TASKS & TEMPLATES, TEMPLATE SETS)

***Arbeiten mit Template Sets***

Auf Workspace (git Intergration) wechseln und in den „FAST/goodTemplate“ Folder wechseln

* Zeigen des Inhalts und der Readme Datei

<right-click folder> and „POST Template SET“

F5 plugin > FAST Menu > TEMPLATE SETs

F5 plugin > FAST Menu > TEMPLATE

*Templates sind hochgeladen und man kann mit denen arbeiten.*

***Deploy & Test Services***

Workspace (GIT) > FAST/goodTemplate Folder > ***README***

* App1 > markieren > <right-click> > ***Deploy Fast App***

Add Virtual Server : 10.1.10.151

Add Node : 212.227.165.10

* App2 > markieren > <right-click> > ***Deploy Fast App***

Add Virtual Server : 10.1.10.152

Add Node : 212.227.165.10

Löschen : <right click> [deployed App] > [delete FAST App]

Löschen : Template Sets

***Rendering Output***

Select App Template in GIT (***app4***) > [***F1***] > F5-Fast: Render HTML Preview

1. **RPM Management**

***Connect to F5-LTM-2***

Install packages: DO, AS3, FAST, TS

<***F1> > F5: Install Update RPM***

*!! Comm-Flow: VScode -> git -> download if not cached -> upload to f5 !!*

Un-Install packages: DO, AS3, FAST, TS

<***F1> > F5: Un-Install RPM***

Workspace > iApp LX Folder > [package installation from git]

* Install DO
* GET Device Info > “DO” eingeben > example

1. **OPEN API stuff …**

Workspace: “***makeHTTPsTesting.md***”

Make anything “API”

Disconnect F5 BigIP

* Cool Feature – “context aware” -> missing host

Mark text > right click > Make HTTP request

***HTTP, YAML, JSON***

**### simple url as string**

```

/mgmt/tm/sys/clock

```

**### simple url in yaml format**

```

url: /mgmt/tm/sys/clock

```

**### simple url in json format**

```

{

  "url": "/mgmt/tm/sys/clock"

}

```

**### url post in yaml**

```

url: /mgmt/shared/authn/login

method: POST

body:

    username: admin

    password: f5twister!

```

**### url post in yaml**

```

url: /mgmt/tm/sys/config

method: POST

body:

    command: save

```

***External Calls***

**### Simple get**

```

https://api.chucknorris.io/jokes/random

```

**### simple get in yaml**

```

url: https://api.chucknorris.io/jokes/random

```

**### simple get in yaml - shows broken destination**

```

url: https://broken.extra.io/whah

```

**### simple POST in yaml**

```

url: 'https://postman-echo.com/post'

method: POST

data:

  hi: yo

```

**### simple POST in json**

```

{

    "url": "https://postman-echo.com/post",

    "method": "POST",

    "data": {

        "hi": "yo"

    }

}

```

**### raw get to external F5 using basic auth - DEV**

```

{

  "url": "https://192.168.200.131/mgmt/tm/sys/ntp",

  "auth": {

      "username": "admin",

      "password": "yayPassword!",

      "sendImmediately": "true"

  }

}

```

1. **Convert // JSON <-> YAML & base64 de- encode**

***VS code extension***: YAML

Settings: YAML schemas -> f5

Workspace (git) > YAML Folder > \*.yaml -> show schema validation

* Hover over
* Add wrong text

Workspace (git) > YAML/NAP Folder > Show NAP syntax

* Convert to YAML

Workspace (git) > base64 Folder > Show “cert import”

1. **Troubleshooting**

Falschen Call erstellen > abschiessen

ID merken

ID suchen