

API

Workshop

J&M GmbH

Ziel: technisch tiefergreifendes Verständnis für HTTP API's mit JSON

- Methoden der Anfrage
- Aufbau von Anfragen (Requests)
- Analyse von Request / Responses im Browser
- Entwicklung eines kleinen PHP-Servers
- Abfragen an den PHP-Server mit eigenem Client
- Aufbau JSON in Verbindung mit OpenAPI
- Umgang mit Langläufern

Voraussetzungen

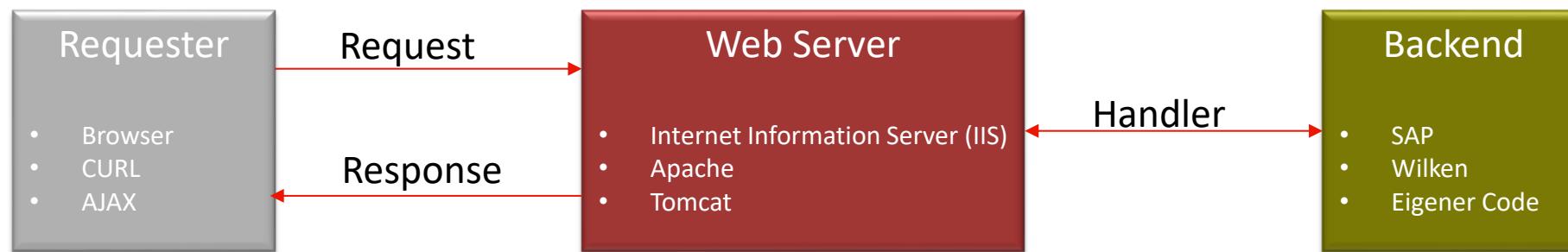
Ziel des Workshops ist es, dass jeder Teilnehmer in der Lage ist, selbstständig einen eigenen WEB-Dienst mittels PHP zu entwickeln, um API's en Detail zu verstehen

- Der Web-Dienst soll zur Demonstration dienen
- Es ist kein Programmierkurs – sondern soll bestimmte Voraussetzungen und Zusammenhänge für API's verdeutlichen
- Laden Sie bitte von <https://makohub.de> im Menü Doku die PHP Installation herunter und entpacken Sie diese in ein von Ihnen gewähltes Verzeichnis. (Login: VornameNachname, PW: makohub, z. B. EdgarJung)
- Testen Sie bitte die Lauffähigkeit des PHP Interpreters durch Eingabe (Windows Eingabeaufforderung): php –version (Sie sollten keinen Fehler sondern eine Versionsinformation erhalten. Bitte nicht Doppelklick auf PHP sondern manuell die Eingabeaufforderung öffnen, in das Verzeichnis wechseln und das Kommando php –version aufrufen)

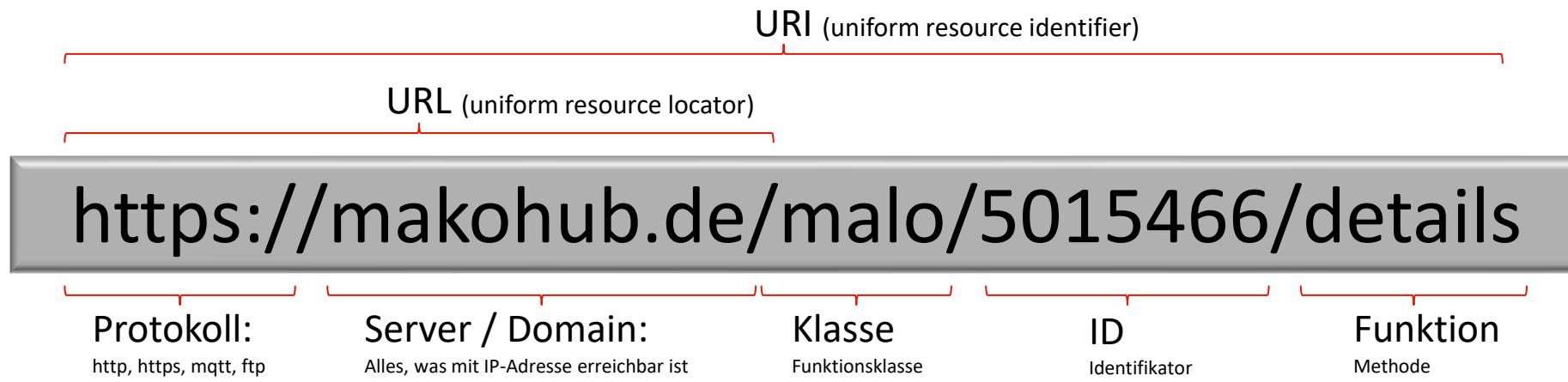
```
C:\Users\EdgarJung>php -version
PHP 8.3.19 (cli) (built: Mar 12 2025 14:03:36) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.3.19, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.19, Copyright (c), by Zend Technologies
    with Xdebug v3.4.2, Copyright (c) 2002-2025, by Derick Rethans
```

- Installieren Sie den Visual Studio Code Editor ([Download Visual Studio Code - Mac, Linux, Windows](#)). Sollten Sie keine Berechtigung für die Installation von Software haben, können Sie alternativ den vorhandenen notepad Editor oder die editor APP von Windows verwenden.

Grundsätzlicher Aufbau von Web API Anwendungen. Strikte Trennung von WebServer (Transportschicht) und Backend (Verarbeitungsschicht)



Aufbau von Requests – hier der URI



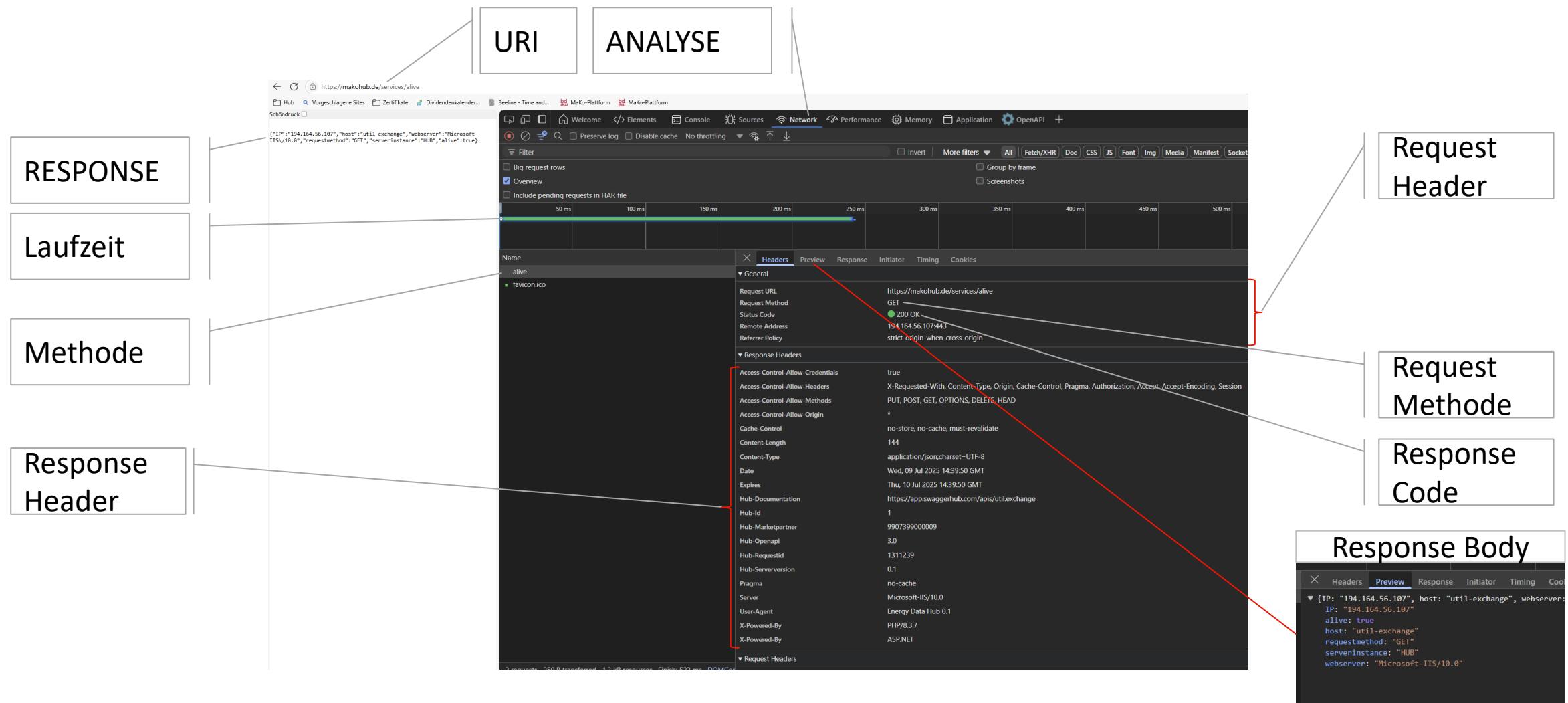
| Beispiel | Bewertung | Info |
|------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| https://makohub.de/malo/v1/ident | DON'T | Moderne Schnittstellen verwenden auf keinen Fall die Version in der URI. Die Version wird im HTTP Header mitgeführt. (Gute Schnittstellen ändern keine Links – siehe TU Dortmund URL-Segment - TYPO3-Doku - TU Dortmund) |
| https://makohub.de/rest.php?malo=5015466 | DON'T | Keine Abstraktion von der dahinterliegenden Implementierung. Es wird in der SSt PHP erwartet. Sie API kann nicht einfach ausgetauscht werden. Die Verwendung von ? Ist technisch möglich, jedoch veraltet |
| https://makohub.de/malo/list | OK | Hier möchte man eine Liste von Ergebnissen erhalten. |
| https://makohub.de/malo/5015466 | OK | OK, wenn die Requestmethode sauber ausgeführt wurde (PUT, GET, POST, DELETE) |

Request Methoden

| Methode | Wirkung | Info |
|----------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GET | Daten lesen | Nur Daten lesen, keine Verarbeitung. Browser Standard. Nur Request URI, Body technisch möglich, könnte jedoch abgelehnt werden. Daher keine Body Daten für GET verwenden |
| POST | Neue Daten anlegen | Nahezu keine Größenbeschränkung. Grundsätzlich Standardmethode in API's |
| PUT | UPDATE komplett | Update von kompletten Datensätzen bzw. UPLOAD von Dateien. Achtung gesonderte Freischaltung am Webserver |
| PATCH | UPDATE Teilweise | Anders wie bei PUT, können hier Teile von Daten ersetzt werden ohne einen kompletten Datenversand erneut zu initiieren. Achtung gesonderte Freischaltung am Webserver |
| DELETE | LÖSCHEN | Löschen von Daten. Muss für REST gesondert freigeschaltet werden. |
| CONNECT | Verbindungsaufbau | Login und Initiierung einer Session. Gesonderte Freischaltung am Server |

Analyse eines Request (IE, Taste F12)

<https://makohub.de/services/alive>



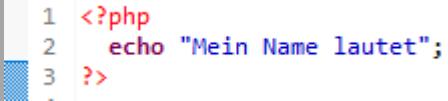
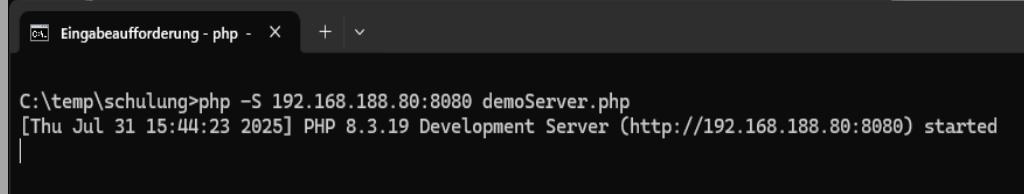
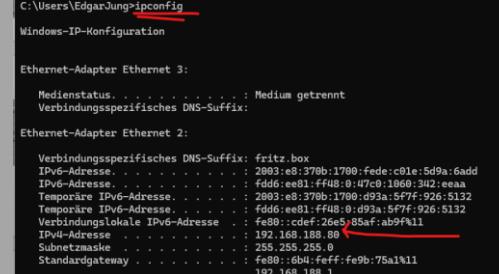
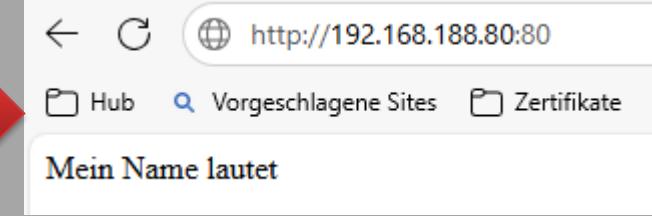
Analysieren Sie

| Nummer | Info |
|--------|---------------------------------------------------------------------------------------|
| 1 | https://makohub.de/services/alive |
| 2 | https://makohub.de/services/running |
| 3 | Loggen Sie sich bei makohub.de aus und testen Sie oben genannte Ergebnisse nochmals |

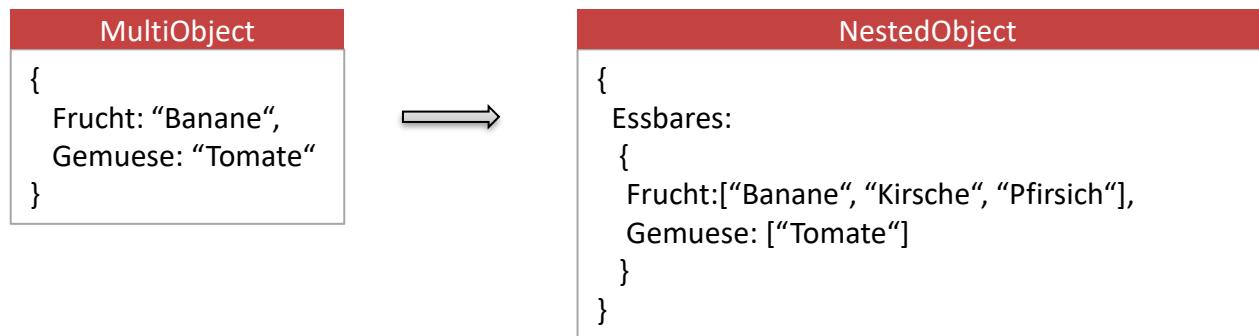
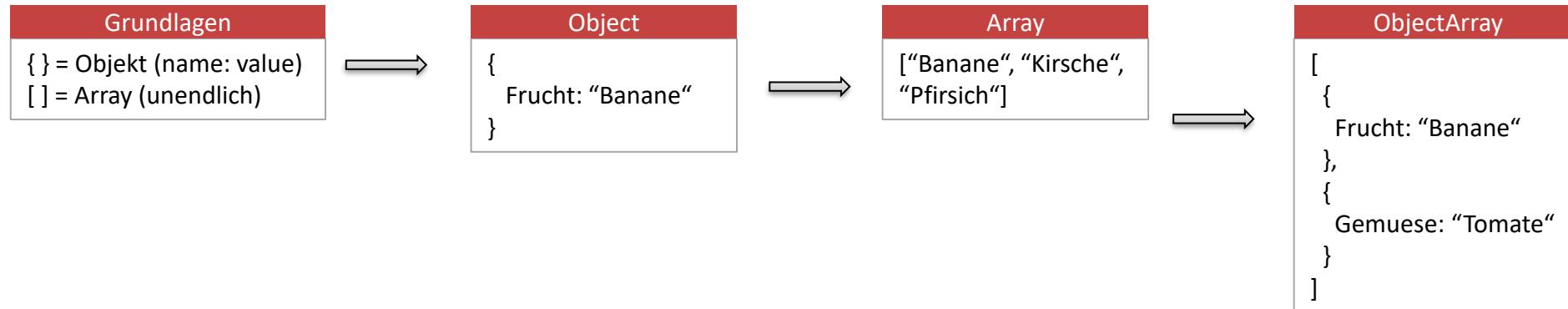
Fehlercodes / http Status Codes

| CODE | Info |
|-----------|---------------------------------------------------------------------------------------------------------------------------|
| 1X | Anfrage läuft noch – wird kaum verwendet |
| 2X | Anfrage war erfolgreich |
| 3X | Anfrage wurde weitergeleitet – hier müssen wir etwas unternehmen, um anschließend einen 2X zu erhalten. Kaum verwendet |
| <u>4X</u> | Der Client hat einen Fehler gemacht (Autorsierung fehlt, Syntax beim Request falsch, oder bei sofortiger Semantikprüfung) |
| 5X | Der Server hat einen Fehler ausgelöst (Meist Programmier- oder Ressourcenfehler [Timeout, Speicher]) |

Entwicklung des ersten eigenen API-Servers mit PHP

| Task | Info |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Schreiben Sie Ihren Code und speichern Sie diesen in das PHP Verzeichnis in die Datei demoServer.php |
| |  |
| 2 | Starten Sie den Server mit php -S <ihreIP>192.168.188.80:80 demoServer.php (Ihre IP-Adresse erhalten Sie mit ipconfig, siehe IPv4Adresse) |
| |  |
| 3 | Rufen Sie im Browser auf: http://192.168.188.80 |
| |  |
| 4 | Lassen Sie sich die IP-Adresse von Ihrem Nachbarn geben (geben Sie ipconfig auf in der Eingabeaufforderung ein). Versuchen Sie den Server Ihres Kollegen zu erreichen: http://192.168.188.80:80 |
| |   |

Strukturierung von Request/Response Werten (Body) in JSON (JavaScript Object Notation)



Strukturierung von Request/Response Werten (Body) in JSON (JavaScript Object Notation)

| Object |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| { Frucht: "Banane" } header("Content-Type: application/json; charset=UTF-8"); \$object1["Frucht"] = "Banane"; echo "object: " .json_encode(\$object1)." ".PHP_EOL; |

| ObjectArray |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [{ Frucht: "Banane" }, { Gemuese: "Tomate" }] header("Content-Type: application/json; charset=UTF-8"); \$object1["Frucht"] = "Banane"; \$object2["Gemuese"] = "Tomate"; \$objectArray[] = \$object1; \$objectArray[] = \$object2; echo "objectArray:".json_encode(\$objectArray)." ".PHP_EOL; |

| Array |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ["Banane", "Kirsche", "Pfirsich"] header("Content-Type: application/json; charset=UTF-8"); \$array = array("Banane", "Kirsche", "Pfirsich"); echo "array: " .json_encode(\$array)." ".PHP_EOL; |

| NestedObject |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| { Essbares: { Frucht:["Banane", "Kirsche", "Pfirsich"], Gemuese: ["Tomate"] } } header("Content-Type: application/json; charset=UTF-8"); \$object1["Frucht"] = "Banane"; \$object2["Gemuese"] = "Tomate"; \$array = array("Banane", "Kirsche", "Pfirsich"); \$nested1["Frucht"] = \$array; \$nested2 = \$object2; \$nestedObject["Essbares"] = \$nested1+\$nested2; echo "nestedObject:".json_encode(\$nestedObject)." ".PHP_EOL; |

| Kompletter Server Code |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <?php header("Content-Type: application/json; charset=UTF-8"); echo "Mein Name lautet Stefan Seidel "; \$object1["Frucht"] = "Banane"; \$object2["Gemuese"] = "Tomate"; \$array = array("Banane", "Kirsche", "Pfirsich"); \$objectArray[] = \$object1; \$objectArray[] = \$object2; \$multiObject = \$object1+\$object2; \$nested1["Frucht"] = \$array; \$nested2 = \$object2; \$nestedObject["Essbares"] = \$nested1+\$nested2; echo "object: " .json_encode(\$object1)." ".PHP_EOL; echo "array: " .json_encode(\$array)." ".PHP_EOL; echo "objectArray:".json_encode(\$objectArray)." ".PHP_EOL; echo "multiObject:".json_encode(\$multiObject)." ".PHP_EOL; echo "nestedObject:".json_encode(\$nestedObject)." ".PHP_EOL; ?> |

Ausführen mit: php -S localhost:8080 demoServer.php

Werte anstatt im Browser, in einem Programm empfangen. Request + Response

Kompletter Server Code (demoCurlClient.php)

```
<?php

// Initialize cURL session
$ch = curl_init();

// Set the URL to fetch
curl_setopt($ch, CURLOPT_URL, "http://localhost:8080");

// Return the transfer as a string instead of outputting it
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// Execute the cURL session
$response = curl_exec($ch);

// Check for errors
if (curl_errno($ch)) {
    echo 'CURLOPT Error: ' . curl_error($ch);
} else {
    // Output the response
    echo $response;
}

// Close the cURL session
curl_close($ch);
?>
```

Reiner API Code (ohne Beschreibung, demoServer.php)

```
[
  {
    Frucht: "Banane"
  },
  {
    Gemuese: "Tomate"
  }
]
```

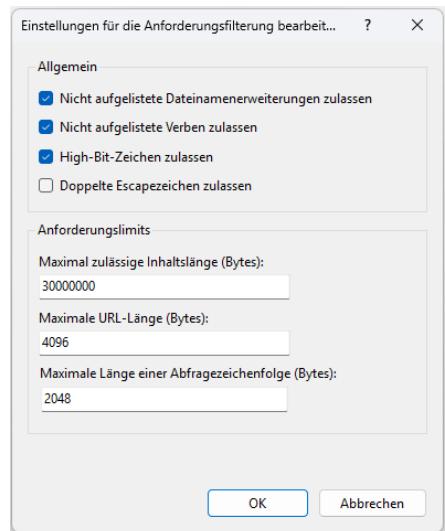
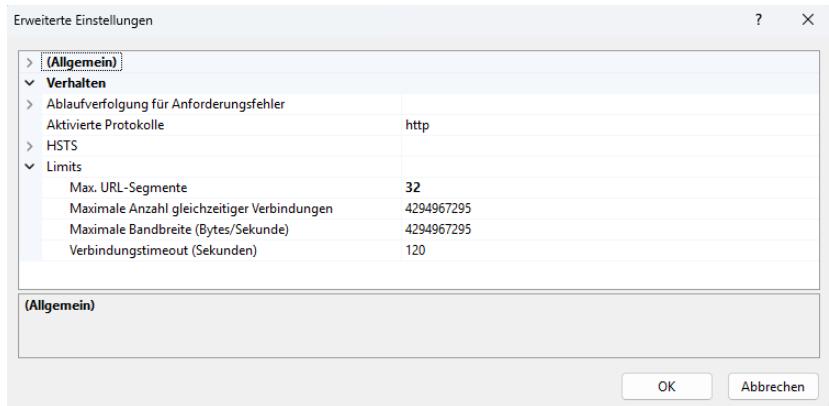
```
header("Content-Type: application/json; charset=UTF-8");
$object1["Frucht"]  ="Banane";
$object2["Gemuese"] = "Tomate";
$objectArray[]       = $object1;
$objectArray[]       = $object2;

echo json_encode($objectArray);
```

Ausführen mit: php demoCURLClient.php

Ausführen mit: php -S localhost:8080 demoServer.php

Einschränkungen, Limits



PHP Einstellungen

Konfiguriere alle Einstellung der Datei php.ini. Diese Einstellungen steuern alle Aspekte

| Name | Wert | Sektion |
|--------------------------------|-------------------------------|---------|
| cgi.fix_pathinfo | 1 | PHP |
| cgi.force_redirect | 0 | PHP |
| default_charset | "UTF-8" | PHP |
| default_mimetype | "text/html" | PHP |
| default_socket_timeout | 60 | PHP |
| disable_classes | | PHP |
| disable_functions | | PHP |
| display_errors | Off | PHP |
| display_startup_errors | Off | PHP |
| doc_root | | PHP |
| enable_dl | Off | PHP |
| engine | On | PHP |
| error_log | "C:\Windows\Temp\php-8.3.... | PHP |
| error_reporting | E_ALL & ~E_DEPRECATED | PHP |
| expose_php | On | PHP |
| extension_dir | "C:\Program Files\PHP\PHP8... | PHP |
| fastcgi.impersonate | 1 | PHP |
| fastcgi.logging | 0 | PHP |
| file_uploads | On | PHP |
| html_errors | Off | PHP |
| ignore_repeated_errors | Off | PHP |
| ignore_repeated_source | Off | PHP |
| implicit_flush | Off | PHP |
| include_path | ".;./;././;./" | PHP |
| log_errors | On | PHP |
| max_execution_time | 800 | PHP |
| max_file_uploads | 20 | PHP |
| max_input_time | 60 | PHP |
| memory_limit | 128M | PHP |
| output_buffering | 4096 | PHP |
| post_max_size | 16M | PHP |
| precision | 14 | PHP |
| register_argc_argv | Off | PHP |
| report_memleaks | On | PHP |
| request_order | "GP" | PHP |
| serialize_precision | -1 | PHP |
| short_open_tag | Off | PHP |
| unserialize_callback_func | | PHP |
| upload_max_filesize | 2M | PHP |
| upload_tmp_dir | "C:\Windows\Temp\" | PHP |
| user_dir | | PHP |
| variables_order | "GPCS" | PHP |
| zend.enable_gc | On | PHP |
| zend.exception_ignore_args | On | PHP |
| zend.exception_string_param... | 0 | PHP |
| zlib.output_compression | Off | PHP |

Problem der Fehlerbehandlung – zu wenige Status Codes. Keine Aussagekraft

Response Envelope und Data

```
{  
  Status: "ERROR",  
  Message: "MaLo did not match",  
  Reference: "See EDI Process Document N513.7",  
  Backend:  
    {  
      Runtime: "Development",  
      EDIVersion: "0.1",  
      Module: "Semantic Check",  
    },  
  Data:  
    {  
      ...  
    },  
}
```

Json Schema

Entire set of control and content

Status string required
Even if we've got 2X from API, there might be a nested Error to handle at client
Allowed values: ERROR WARNING OK
Example: ERROR

Message string required
Message in Envelope Header
Example: MaLo did not match

Reference string
Description, what happened for improved error analysis
Example: See EDI Process Document N513.7

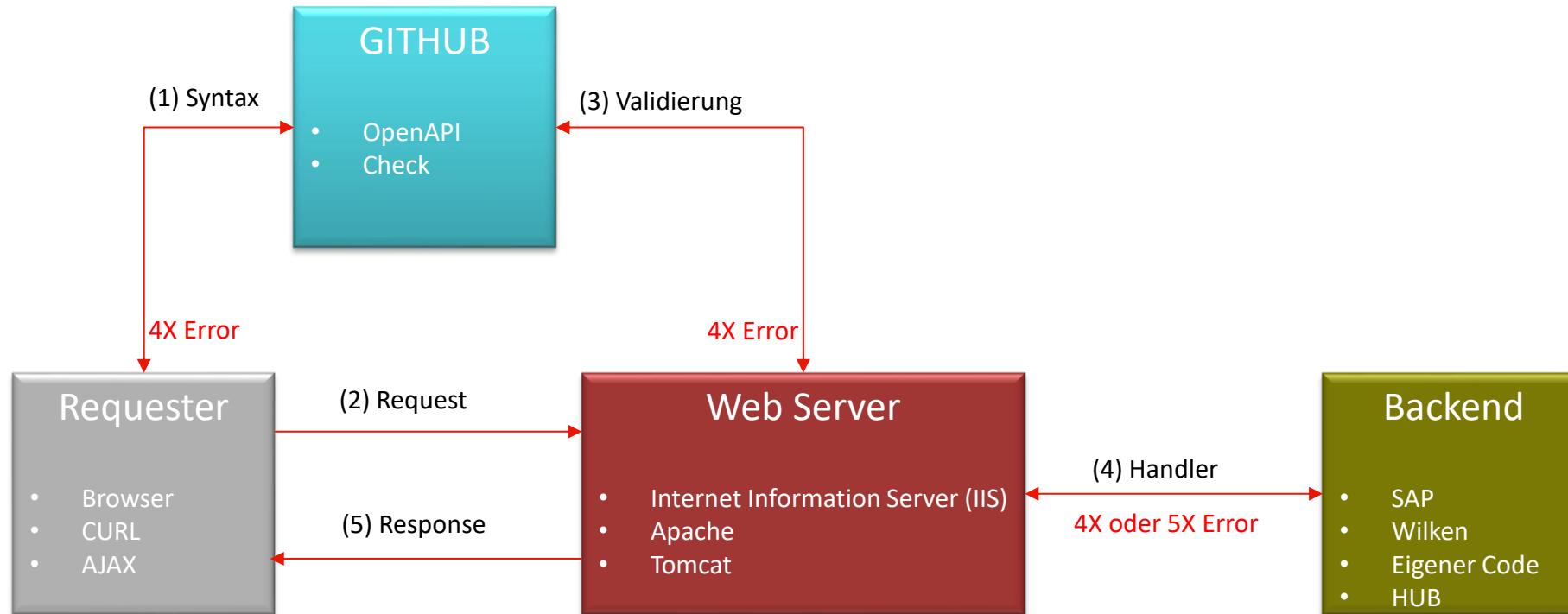
Backend object

- **Runtime** string required
Type of System
Allowed values: Development Quality Production
Example: Development
- **EDIVersion** string required
Version of Interface, consulted by EDI@ENERGY
Match pattern: \d\d\d
- **Module** string required
Hint, which Module is affected

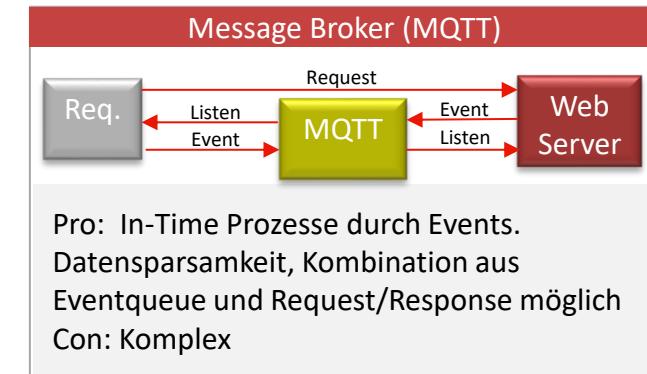
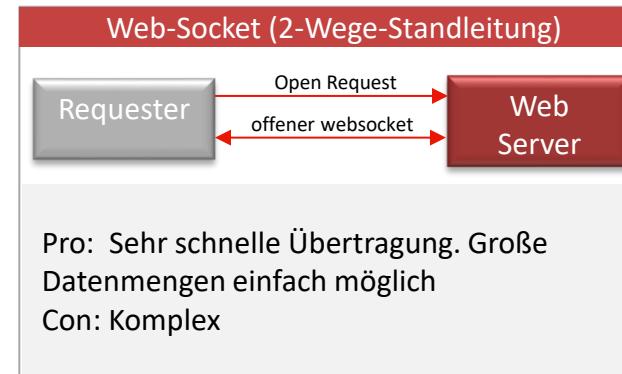
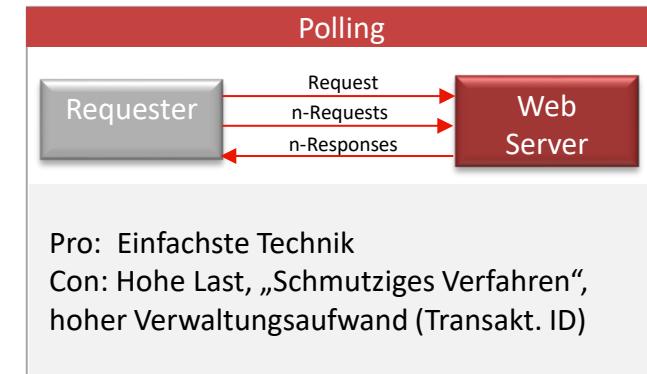
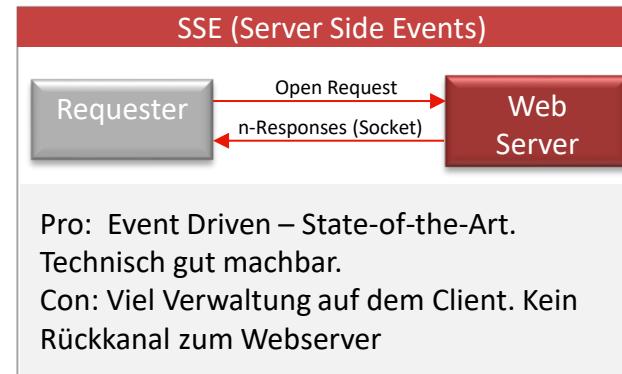
Data object
Real Data to transport

```
type: object  
properties:  
  Status:  
    $ref: Status.yaml  
  Message:  
    $ref: Message.yaml  
  Reference:  
    $ref: Reference.yaml  
  Backend:  
    $ref: Backend.yaml  
  Data:  
    type: object  
    properties: {}  
    description: Real Data to transport  
required:  
  - Status  
  - Message  
description: Entire set of control and content
```

Art der Fehlerbehandlung (Syntax, Semantik)



Umgang mit Langläufern (grundsätzlich 30 sec bis 120 sec zwischen Request und Response, sonst Timeout 408, 504)



Vielen Dank für Ihre
Aufmerksamkeit