# Aufgabe 2

Jede Gruppe soll für ihr Anwendungsbeispiel in einem Worddokument

* den verwendeten Stack beschreiben
* Die verwendeten Frameworks benennen und
* Die benötigten REST Services und dazugehörigen Ressourcen beschreiben

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | https://openclipart.org/image/2400px/svg_to_png/3677/torfnase-Arrow-left-right.png | http://www.rudeprogrammer.com/wp-content/uploads/2016/01/entity-framework1.png |
|  | | | http://www.asphostcentral.com/Images/logo_aspmvc345.png |
| Ähnliches Foto | ScanParkDB | | http://www.logospike.com/wp-content/uploads/2016/05/Asp.net_Logo_03.png |
| Ähnliches Foto | | | https://upload.wikimedia.org/wikipedia/de/1/15/IIeSs-8.png |
| https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Windows-server-2016.png/320px-Windows-server-2016.png | | | https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Windows-server-2016.png/320px-Windows-server-2016.png |
|  | | |  |
| DB-Server | | | APP-Server |
| https://openclipart.org/image/2400px/svg_to_png/3677/torfnase-Arrow-left-right.png | | | | |
| http://www.springpeople.com/media/catalog/product/l/e/learn-javascript.png | | ES6-promise, Ladda Bootsptrap, Material-Kit, Sweet Alert 2 | | |
| https://datatables.net/media/images/nav-dt.png | | DataTables.net | | |
|  | | Bootstrap | | |
| http://www.edututorials.com/Content/images/Tutorial/JQuery/JQuery-icon.png | | jQuery | | |
| https://lh3.googleusercontent.com/nYhPnY2I-e9rpqnid9u9aAODz4C04OycEGxqHG5vxFnA35OGmLMrrUmhM9eaHKJ7liB-=w300 https://maxcdn.icons8.com/Share/icon/Logos/ms_edge1600.png http://findicons.com/files/icons/783/mozilla_pack/256/firefox.png https://upload.wikimedia.org/wikipedia/commons/thumb/4/49/Opera_2015_icon.svg/1024px-Opera_2015_icon.svg.png | | | | |
| Betriebssystem | | | | |
|  | | | | |
| Client | | | | |

REST Web-Service URLs

## Rechnungen auslesen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Bill/Bill | GET | Gibt eine Liste mit allen Rechnungen für den aktuell angemeldeten Benutzer zurück (bezahlte und unbezahlte Rechnungen) | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {} | | | [{  "carPark" : 329,  "licensePlate" : "KA XX 1234",  "date" : "14.03.2017",  "entrence" : "10:27",  "exit" : "11:45",  "duration" : "01:18",  "amount" : "3,50",  "paied" : false,  "billId" : 5  }] | |

## Rechnungen erstellen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Bill/Bill | POST | Erstellt eine neue Rechnung für das ermittelte Kennzeichen und erhält als Antwort den Status. Diese URL wird vom Parkhaus aufgerufen. | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {  "carPark" : 329,  "licensePlate" : "KA XX 1234",  "date" : "14.03.2017",  "entrence" : "10:27",  "exit" : "11:45",  "amount" : "350"  } | | | {  "status" : "success"  } | |

## Rechnungen bezahlen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Bill/Bill | PUT | URL wird aufgerufen, sobald der Benutzer auf „Bezahlen“ klickt. | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {  "billId" : 2  } | | | {  "status" : "success"  } | |

## Fahrzeugliste auslesen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Car/Car | GET | URL wird aufgerufen, sobald der Benutzer auf „Bezahlen“ klickt. | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {} | | | [{  "registeredPlateId":42,  "profileId":1,  "plateDistrict":"ED",  "plateIdentifier":"KA",  "plateNumber":1000,  "brand":"VW",  "model":"Golf"  }] | |

## Fahrzeug anlegen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Car/Car | POST | URL wird aufgerufen, sobald der Benutzer auf „Fahrzeug hinzufügen“ klickt. | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {  "profileId":1,  "plateDistrict":"ED",  "plateIdentifier":"KA",  "plateNumber":1000,  "brand":"VW",  "model":"Golf"  } | | | {  "status" : "success"  } | |

## Fahrzeug bearbeiten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Car/Car | PUT | URL wird aufgerufen, sobald der Benutzer auf „Fahrzeug bearbeiten“ klickt. | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {  "registeredPlateId":42,  "plateDistrict":"ED",  "plateIdentifier":"KA",  "plateNumber":1000,  "brand":"VW",  "model":"Golf"  } | | | {  "status" : "success"  } | |

## Fahrzeug löschen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ressource** | **Methode** | **Erwartete Ausgabe** | | **Response Code** |
| /Car/Car | DELETE | URL wird aufgerufen, sobald der Benutzer auf „Fahrzeug löschen“ klickt. | | 200 / OK |
|  |  |  | |  |
| **Daten Request** | | | **Daten Response** | |
| {  "registeredPlateId":42  } | | | {  "status" : "success"  } | |