

Dokumentation der POC

POC1 Trainingsplan erstellung

- **Welches Risiko adressiert der PoC?**

Trainingsplanerstellung aus verschiedenen Modulen, die aus der Datenbank gezogen werden

- **Was ist das Ziel des PoC?**

Die Kernfunktion (einen Trainingsplan erstellen) zu gewährleisten

- **Exit Kriterium**

Ein korrekter Trainingsplan wird erstellt aus den Daten, die im Kostmodus eingegeben werden

Fail Kriterium

Kein Trainingsplan wird dargestellt

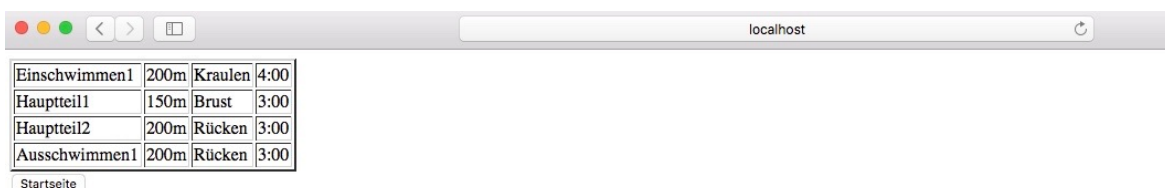
- **mögliche Alternativlösungen (Fallbacks) falls PoC nicht erfüllt werden kann**

Der Trainingsplan wird aus Daten aus der Datenbank erstellt.

- **geplanter Termin und Status**

06.11.15 Trainingsplan wird aus eingegebenen Modulen erstellt.

12.11.15 Trainingsplan wird aus zufällig gewählten Modulen erstellt
erfüllt jedoch die Kriterien eines kompletten Trainingsplans



The screenshot shows a web browser window with the address bar displaying 'localhost'. Below the browser window, there is a table with 4 columns and 4 rows of data. The table is titled 'Startseite' in the bottom left corner.

Einschwimmen1	200m	Kraulen	4:00
Hauptteil1	150m	Brust	3:00
Hauptteil2	200m	Rücken	3:00
Ausschwimmen1	200m	Rücken	3:00

Die GET-Methode **/modulliste** in app.js wird in trainingsplan.html aufgerufen, um auf die Ressource „Modulliste“ zuzugreifen. Als erstes wird die Tabelle *Modul* an die Datenbank *db* gebunden, um mit der Variable *daba* weiter zu arbeiten. Anschließend erfasst **daba.findItems(function(error, result) {})** alle Datensätze, speichert sie in *result* ab und gibt sie mit **res.end(JSON.stringify(result))** im JSON-Format an trainingsplan.html zurück, in der die Daten weiter verarbeitet werden.

app.js:

```
app.get('/Modulliste', function (req, res, next)
{
    db.bind("Modul");
    var daba = db.Modul;

    daba.findItems(function(error, result) {
        if(error)
            next(error);
        else {
            res.writeHead(200, {
                'Content-Type': 'application/json'
            });
            res.end(JSON.stringify(result));
        }
    });
});
```

Als erstes wird eine *request* Variable initialisiert, worauf der ajax-Befehl ausgeführt wird. Dieser beinhaltet die GET-Methode **/Modulliste**, wo alle Datensätze nach den Kriterien (in diesem Beispiel: findItems = alle Datensätze) abgerufen werden. Beim erfolgreichen ausführen werden diese mit **request.done(function(data) {})** in *data* gespeichert. In **data.forEach(function(team) {})** werden die einzelnen Module an **addTableRow(Modul)** übergeben, wo sie in der Tabelle (id: *tabelle*) mit **.append** hinzugefügt werden. Wenn jedoch das Abrufen der Datensätze von **/Modulliste** fehlgeschlagen ist, ruft er in **request.fail(function(error) {})** mit **alert()** eine Fehlermeldung auf.

```
$(document).ready(function() {

    event.preventDefault();

    var request = $.ajax({
        type: 'GET',
        url: '/Modulliste',
        contentType: 'application/json'
    })
    request.done(function(data) {
        data.forEach(function(modul) {
            addTableRow(modul);
            console.log(modul);
        });
    })
    request.fail(function(error) {
        alert('Es ist ein Fehler aufgetreten.');
```

```

    });

    $('#home').on('click', function () {
        window.location.replace('index.html');
    });
});

var client = new Faye.Client('/faye');
var subscription = client.subscribe('/modul',
function(msg) {
    addTableRow(msg);
});

function addTableRow(modul){
    $('#table#tabelle tbody').first().append
    ('<tr><td>'+modul.name+'</td><td>'+modul.strecke+'</td><td>'+modul.lage+'</td><td>'+modul.zeit+'</td></tr>');
};

```

POC2 GPS-Uhr

▪ Welches Risiko adressiert der PoC?

Übertragung der Pulsdaten der GPS Uhr von dem letzten Training

▪ Was ist das Ziel des PoC?

Die Pulsdaten der GPS Uhr sollen an die App übermittelt werden, damit das letzte Training ausgewertet werden kann.

▪ Exit Kriterium

Von 10 Trainingseinheiten werden die Pulsdaten an die App übermittelt und die Daten können ausgewertet werden.

Fail Kriterium

Keine Daten werden an die App von der GPS Uhr übermittelt

▪ mögliche Alternativlösungen (Fallbacks) falls PoC nicht erfüllt werden kann

Die Pulsdaten werden von der GPS Uhr abgelesen und manuell eingegeben

▪ geplanter Termin und Status

30.11.15 Exit Kriterium wird erfüllt

Das POC GPS-Uhr ist zu diesem Zeitpunkt im Fail Status, weil noch keine GPS-Uhr zur Testung zu Verfügung steht. Diese wird erst in der nächsten Woche von dem Domänenexperte besorgt. Die Funktionen der Uhr sollen über das Connect IQ SDK verbunden werden.

Garmin Swim <https://buy.garmin.com/de-DE/DE/uhren/armbänder/garmin-swim-/prod97287.html> wäre eine Schwimmuhr, die für die SwimFit App sinnvoll wäre. Sie zeichnet Distanz, Pace, Anzahl der Züge und Schwimmstil auf. Sie ist fähig drahtlos die Daten zu übertragen mit der ANT+™-Technologie.

POC 3 Trainingsauswertung

▪ Welches Risiko adressiert der PoC?

Darstellung der Trainingsdaten in einer Übersichtlichen Statistik

▪ Was ist das Ziel des PoC?

Die Pulsdaten sollen in einem Diagramm die Trainingsentwicklung darstellen.

▪ Exit Kriterium

Pulsdaten werden in einem Diagramm dargestellt.

Fail Kriterium

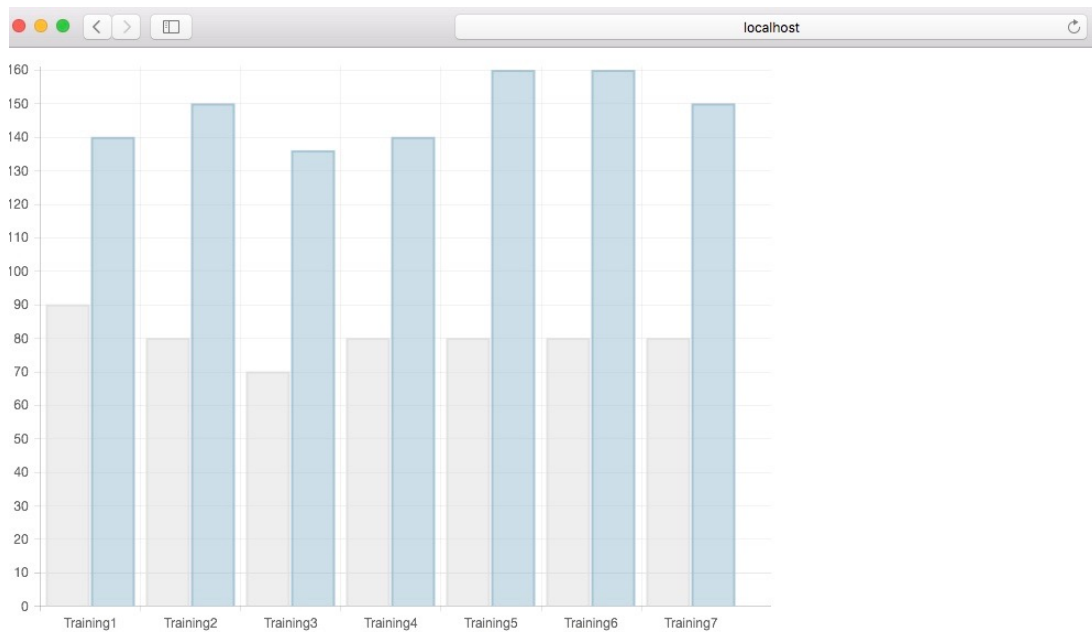
Keine Daten werden in einem Diagramm dargestellt

▪ mögliche Alternativlösungen (Fallbacks) falls PoC nicht erfüllt werden kann

Die Pulsdaten werden in einer Tabelle ausgegeben

▪ geplanter Termin und Status

06.11.15 Exit Kriterium wird erfüllt.



Für die Darstellung der Trainingspulsdaten in einem Diagramm wird die Chart.js-Bibliothek verwendet.

In Hellgrau ist der Ruhepuls abgebildet und in Hellblau der Belastungspuls. Datensätze aus 7 verschiedenen Trainings werden dargestellt und sind somit visuell vergleichbar.

Der Screenshot zeigt an einem Beispiel, wie ein Diagramm in eine Seite eingebunden wird.

```

<!doctype html>
<html>
  <head>
    <title>Pie Chart</title>
    <script src="../Chart.js"></script>
  </head>
  <body>
    <div id="canvas-holder">
      <canvas id="chart-area" width="300" height="300"/>
    </div>

    <script>
      var pieData = [
        {
          value: 300,
          color: "#F7464A",
          highlight: "#FF5A5E",
          label: "Red"
        },
        {
          value: 50,
          color: "#468FBD",
          highlight: "#5AD3D1",
          label: "Green"
        },
        {
          value: 100,
          color: "#FDB45C",
          highlight: "#FFC870",
          label: "Yellow"
        },
        {
          value: 40,
          color: "#949FB1",
          highlight: "#A8B3C5",
          label: "Grey"
        },
        {
          value: 120,
          color: "#4D5360",
          highlight: "#616774",
          label: "Dark Grey"
        }
      ];

      window.onload = function(){
        var ctx = document.getElementById("chart-area").getContext("2d");
        window.myPie = new Chart(ctx).Pie(pieData);
      };
    </script>
  </body>
</html>

```