



703128 PS/2 Web Services  
**Project Final Report**

**Team Name :** HATI (Hiking and Transportation in Innsbruck)

No.	Matrikel Nr.	Name
1	1316136	Bernhard Fritz
2	1318235	Thibault Gerrier
3	1217362	Raphael Gruber
4	1315434	Philipp Häusle
5	1315417	Mario Zelger

## Project Pitch

Define the idea of your project in one sentence.

The idea of HATI is a web application which offers you possible hiking routes around Innsbruck also including public transport to get to the starting point of the routes.

## Project Use Case

Define the potential use cases for your project, for example the cases where your system could be applied.

When you are in Innsbruck as a tourist and plan to do a hiking trip, HATI is the perfect solution:

- **Search for possible hiking routes**  
With HATI you can search for tested hiking routes all around Innsbruck. All routes have been created and commented from outdooractive-users. There are also pictures available for most of the routes.
- **Take a look at the selected route drew on a map**  
The selected route will be visible on an OpenStreetMap so the user can take a look at it before (s)he starts the trip.

- **Use public transport to get to the starting point of the route**  
HATI offers the user a public transport solution to get to the starting point of the route. The necessary bus stops are also drawn on the map.
- **Get the weather forecast for the area of the hiking route**  
For a selected hiking route HATI shows the user the weather forecast for the next five days.
- **Visualize the current user position during the trip**  
HATI allows the user to visualize his/her current position on the map to check it with the drawn hiking route.

Notice: The target audience of HATI are tourists during their holidays in Innsbruck. But also local users can use HATI to explore new hiking routes all around Innsbruck.

## Project Detailed Description

Describe in detail your project idea, settings and technical and usage restrictions, limitations (they can be listed using bullet points, providing sufficient details).

The idea of HATI is a web application which offers you possible hiking routes around Innsbruck. The routes are visualized in a map and more information about the routes like length or duration is given. Most of these routes don't start in the center of Innsbruck so you have to get to the start point of the route. For that problem HATI also shows you how to get there using public transport (buses from IVB). As a feature our application also provides weather data in the form of forecasts for the next 5 days, so you can plan your trip accordingly.

HATI will be hosted on [www.heroku.com](http://www.heroku.com) for an easy access.

An example usage of HATI:

1. You are in Innsbruck (e.g. as a tourist) and plan to do a hiking trip.
2. Start HATI and choose one of the offered beautiful hiking routes all around Innsbruck.
3. Check the route informations (also pictures in most of the routes) and confirm your selection.
4. Based on your current position HATI will show you how to get to the start point of the route using public transport.
5. The application also shows the weather forecast for the trip in the area of the selected hiking route.
6. After you have started your trip you are always able to visualize your current position on the map and check it with the hiking route also drawn on the map.
7. Simply follow the route to the end point and enjoy your trip!

HATI combines hiking routes, public transport and weather information to offer beautiful recommended hiking routes around Innsbruck and how to get there easily. The weather forecast is a useful feature to plan the trip accordingly.

HATI is an easy way to get to know the beauty of nature around Innsbruck.

## Project Technical Solution

Describe the technical solution of your project, particularly, depict an architecture and list and describe technical components and interactions among them, used datasets, servers, data structures, databases/repositories, etc. – also provide typical examples.

For this project we decided to use node.js to implement our webserver. Node.js gives you the freedom to organize your project as you would like to. Even if we could write our own webserver using node.js right out of the box, we decided to use express.js, a npm (node package manager) package, to make our lives easier. Express.js gives you the opportunity to program in a MVC (model, view, control) like fashion. The good thing about MVC pattern is that it allows you to swap an entire part (either model, view or control) in a later stage of your project. This can be valuable if you realize you made a mistake in the planning stage. Express.js also allows the server to handle requests with ease. A simple JavaScript function will be called if the server receives a request.

Our view consists of a web app which is hosted on our express.js server. This web app is implemented using Jade, a node templating engine. Jade allows us to program dynamic web pages by giving us an interface to share variables between server and client. Once Jade is 'rendered' the client will receive a plain HTML page. Bootstrap is used to make our web app better looking and responsive.

The web app is basically an interface to the server's functionality. The server handles all requests to other APIs like openstreetmap, outdooractive and openweathermap. It can be considered as a wrapper which modifies the data received by APIs accordingly and presents the result to the client in a pleasant way. In it's final state the web app provides functionality to choose from a list of tours or search for a specific one by entering a keyword. These tours are provided by outdooractive API. Once the user found a tour (s)he can check out tour details like description, length of the tour, altitude or duration. In addition to that, the user has the opportunity to visualize the route on openstreetmap. A weather forecast provided by openweathermap API allows the user to plan the trip accordingly.

We made a wrapper for each API that we are using:

- openStreetMap.js:  
Allows us to fetch Relations and Nodes provided by openstreetmap API. Relation can be considered as a collection of related nodes. For example a bus route consists of a relation of bus stops. Each bus stop has a corresponding openstreetmap node which includes valuable information like bus stop name as well as latitude and longitude coordinates. By using the user's geolocation and the bus stop coordinates we are able to guide the user to nearest bus stop.

Since IVB (Innsbrucker Verkehrsbetriebe) does not provide open data we decided to write our own algorithm to determine the shortest path between two busstops. This will be accomplished by using the acquired data from openstreetmap in a graph like data structure and Dijkstra shortest path algorithm for weighted directed graphs.

- **outdooractive.js:**  
Outdooractive offers tons of tours for hikers, as well as bikers all around the world. By using specific requests we were able to filter for hiking tours around Innsbruck. Some tours were way to long in length considered as tour in Innsbruck, though. That's why we decided to only consider tours which are shorter than 15km and require less than 6 hours.
- **openweathermap.js:**  
This wrapper offers functions to retrieve current weather data from openweathermap API by GPS coordinates or by city name. The provided information consists of temperature, humidity, windspeed, atmospheric pressure information as well as basic indicators of weather states like cloudy, rainy, clear sky and so on.

Ultimately this project will be hosted on [www.heroku.com](http://www.heroku.com). Heroku is a free solution to host web applications on the internet.

## Included Services

List and describe all external services you have currently included in your project and for each service, describe the following aspects:

1. What technologies do you currently use to communicate with the service (e.g. HTTP, SOAP, REST)
2. What data format do you currently use to communicate with the service (e.g. XML, JSON)
3. Which programming libraries do you currently use to send requests and analyze the received responses from the service, please be specific (e.g. Jersey, Spring Social, PHP, SOAP)

No.	Name of the used external services	Description e.g. technical details for the service and its use, technologies involved, other remarks
1	Openstreetmap API	<p>1: We use HTTP GET requests to retrieve bus stations from the Openstreetmap REST API</p> <p>2: The Openstreetmap REST API responds with XML which we convert to JSON using a npm package called "xml2json" (<a href="https://www.npmjs.com/package/xml2json">https://www.npmjs.com/package/xml2json</a>). The JSON data can be parsed to a JavaScript object using the JSON.parse() function.</p> <p>3: We use a npm package called "request"</p>

		<p>(<a href="https://www.npmjs.com/package/request">https://www.npmjs.com/package/request</a>)</p> <p>Example request to <a href="http://api.openstreetmap.org">http://api.openstreetmap.org</a>:</p> <p>/api/0.6/relation/2611781</p> <p>Yields information about the bus line O including all bus stops as well as their GPS coordinates.</p>
2	Outdooractive API	<p>1: We use HTTP GET requests to retrieve tours around Innsbruck from the Outdooractive REST API</p> <p>2: The Outdooractive REST API responds with JSON which can be parsed to a JavaScript object using the JSON.parse() function.</p> <p>3: We use a npm package called "request"</p> <p>Example request to <a href="http://www.outdooractive.com">http://www.outdooractive.com</a>:</p> <p>/api/project/alpenvereinaktiv/filter/tour?key=yourtest-outdoora-ctiveapi&amp;area=1022439&amp;category=1566480</p> <p>Yields ids to specific hiking tours around innsbruck. These ids can used to retrieve more information in a seperate request.</p>
3	Openweathermap API	<p>1: We ues HTTP GET requests to retrieve Innsbruck's weather infromation from the Openweathermap REST API</p> <p>2: The Openweathermap REST API responds with JSON which can be parsed to a JavaScript object using the JSON.parse() function.</p> <p>3: We use a npm package called "request"</p> <p>Example request to <a href="http://api.openweathermap.org">http://api.openweathermap.org</a>:</p> <p>/data/2.5/weather?q=Innsbruck&amp;units=metric</p> <p>&amp;appid=2de143494c0b295cca9337e1e96b00e0</p> <p>Yields current weather information for Innsbruck.</p>

4	Yournavigation API	<p>1: We use HTTP GET requests to retrieve route information from the Yournavigation REST API</p> <p>2: The Yournavigation REST API responds with JSON which can be parsed to a JavaScript object using the <code>JSON.parse()</code> function.</p> <p>3: We use a npm package called “request”</p> <p>Example request to <a href="http://www.yournavigation.org">http://www.yournavigation.org</a>:</p> <p><code>/api/1.0/gosmore.php?format=geojson&amp;flat=...&amp;flon=...&amp;tlat=...&amp;tlon=...</code></p> <p>Yields route information from GPS coordinates flat, flon to GPS coordinates tlat, tlon. This is useful for us to guide the user to the next bus stop.</p>
---	--------------------	---

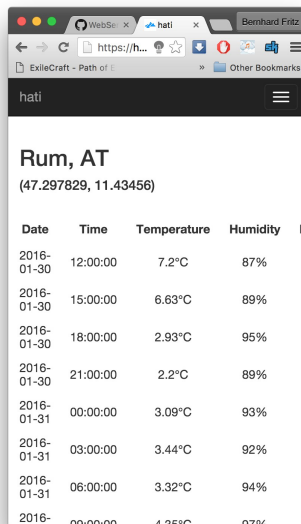
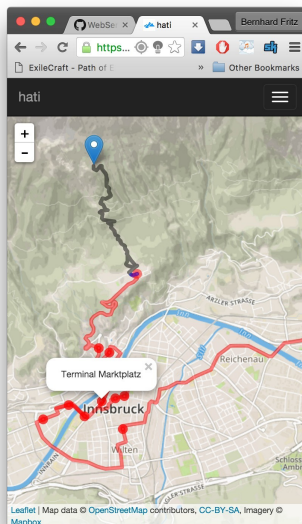
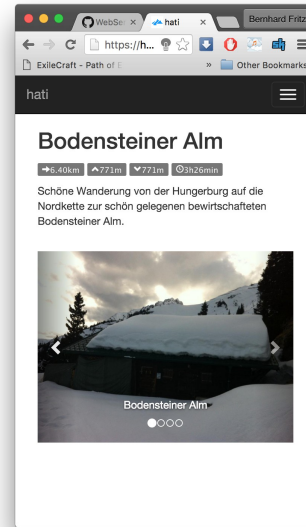
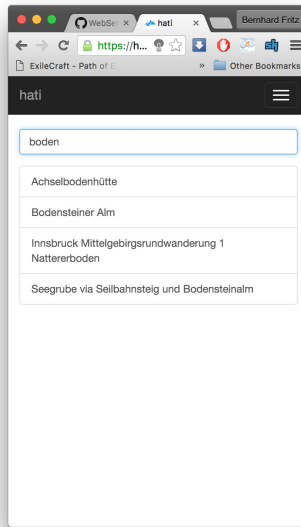
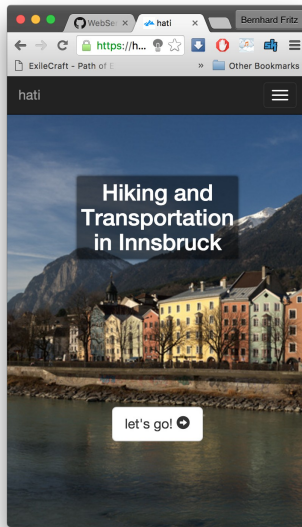
## Result

Describe your final implementation status, including the project's user and machine to machine interfaces (put screenshots if necessary). Also, provide an URI to the resulting service and/or the project code if published online, e.g. as source code.

### Implementation status:

Nearly all our goals have been fulfilled. The only thing left to do is adding busline information to a popup whenever a user clicks on a line in leaflet.

## Project's user interfaces:



## Machine to machine interfaces:

### **GET** /busPoints?from=lat1,lng1&to=lat2,lng2

Finds closest busstops at gps coordinates (lat1, lng1) and (lat2, lng2). These busstops are registered in our IVB bus lines graph. This graph was generated by using data from OpenStreetMap API. To calculate the shortest path between two vertices we used Dijkstra algorithm. The response of the GET request is the path of the bus(es) in correct order encoded in JSON. Example response:

```
[
  [
    47.2641853,
    11.3860903,
    "Klinik/Universität"
  ],
  [
    47.2659964,
    11.3825653,
    "Fürstenweg"
  ],
  ...
]
```

**GET** /reverseGeocoding?lat=...&lon=...

Returns Street, City and also busstop information. We stopped using this function in a later stage of the project since we found an alternative solution. This service is provided by <http://nominatim.openstreetmap.org/>

**GET** /

Returns the main page in HTML

**GET** /search

Returns the page where you can search for hiking tours around Innsbruck.

**GET** /searchresults?title=...

Returns an array containing search results when querying outdooractive with the search term. The data is retrieved by outdooractive API.

Example response:

```
[
  {
    "lineOptions": {
      "strokeColor": "#ff0000"
    },
    "category": {
      "id": "1566480"
    },
    "title": "Achselbodenhütte",
    "geometry": "11.387938,47.267931",
    "id": "105263511",
    "frontendtype": "tour",
    "requestedId": "105263511"
  },
  {
    "lineOptions": {
      "strokeColor": "#ff0000"
    }
  }
]
```



```

    "category": {
      "id": "1566480"
    },
    "title": "Bodensteiner Alm",
    "geometry": "11.396698,47.286249",
    "id": "102453031",
    "frontendtype": "tour",
    "requestedId": "102453031"
  },
  ...
]

```

#### **GET /navigation**

Renders the leaflet map.

#### **GET /leaflet**

Also renders the leaflet map.

#### **GET /tour?id=...**

Returns an HTML page containing detailed information about the chosen hiking tour. The detailed tour information is retrieved by outdooractiveAPI.

#### **GET /points**

Returns the hiking tour gps waypoints as a JSON array. These waypoints are retrieved by the outdooractive API.

#### **GET /route?from=...&to=...&typeOfTransport=...**

Returns an array of gps waypoints to get from A to B by using a specific type of transport (by foot / by car). We use the Yournavigation API to retrieve this data.

#### **GET /weather**

Returns an HTML page with a weather forecast based on the previously picked hiking tour start location. The data is provided by OpenWeatherMap API.

The service can be accessed on heroku: <https://hatiapp.herokuapp.com/>

The source code is on github: <https://github.com/RaphaelHub/WebServices/tree/master/hati>

## **Potential Further Societal and Economic Impacts**

Describe your opinions on the added values of your project from the societal and economic standpoints. For example, how the project can help to create a new business (e.g. startup or a new service in an established company) and what would be necessary for it.

Our idea is to extend the application with a rating system for alps. The alps could pay to get listed in HATI or to get preferred when a user searches for possible hiking routes. The rating system could help smaller alps to become more popular and the payments can be used for continual improvements and new features for HATI.

## Potential Future Work

Describe how you would progress with and/or improve your project, if you are to continue working on it.

The first thing to do is adding busline information to a popup whenever a user clicks on a line in leaflet as mentioned.

An improvement idea is the feature of time limit based hiking route suggestions.

Example: "Where can I get from here in 3 hours?"