Spezifikation eines REST-Webservices

Ranger

* Gefahren, Warnungen
* Besondere Ereignisse (z.B. Neugeborene Tiere, blühende Pflanzen)

Nutzer (Unterteilung in Profis und Laien/Urlauber)

* Schwarzes Brett – z.B. für besondere Ereignisse, oder Gefahren
* Sich in Gruppen eintragen
* Profis dürfen Gruppen erstellen
* Unterkunft buchen
* Kontakt zu Ranger, Notfall
* Bestätigung Anfang/Ende der Tour

Admin

* Gruppen anlegen & allgemeine Infos
* Karte mit allen Infos
* Wetterbericht
* Allgemeine Infos zum Park (Tier, Pflanzen, Routen)
* Unterkünfte
* Events/ besondere Ereignisse
* Bestätigung schwarzes Brett ?

Ressourcen

/Aktivität

anlegen und Informationen zu Aktivitäten (Admin)

/klettern

/Informationen

(auch Routen)

/Gruppen

/wandern

/allgemeine Infos, Routen

/Gruppen

/angeln

/allgemeine Infos, Routen

/Gruppen

eintreten (Nutzer – Laien, Pros),

erstellen (Nutzer – Pros),

austreten (Nutzer)

löschen (Admin /automatisch nach Ablauf des Datums)

/news

/wetterbericht (Admin)

/Gefahren (Ranger)

/Events (Ranger, Admin)

/Unterkunft

/camping

/Ort (Infos)

/Anfrage (Datum, Personenanzahl)

/hütten

/Ort (Infos)

/Anfrage (Datum, Personenanzahl)

/hotel

/Ort (Infos)

/Anfrage (Datum, Personenanzahl)

/kontakt

/Ranger

/Notarzt

/mitteilungen

Kapitel 4 und 5 Stefan TEhlkopf

/ gruppen

Filtern:

Nach aktivitaet, Aktuelleste zuerst anzeigen, nach bestimmtem Datum, freie Plätze, schwierigkeitsgrad, Dauer

Ausrüstung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ressource | Methode | Semantik | Content-type (req) | Content-type  (res) |
| /aktivitaet | POST | Anlegen einer neuen Aktivität (admin) | application/json | application/json |
| /aktivitaet | GET | Abrufen aller Aktivitäten | application/json  application/html  plain | application/json |
| /aktivitaet/klettern | GET | Abrufen der Aktivität |  |  |
| /aktivitaet/klettern /informationen | GET | Abrufen von Informationen, Routen zur Aktivität |  |  |
| /aktivitaet/klettern /gruppen | GET | Abrufen aller Gruppen zu der Aktivität |  |  |
| /aktivitaet/klettern /gruppen | POST | Anlegen einer Gruppe (Nutzer – Pros) |  |  |
| /aktivitaet/klettern /gruppen/ID=1 | PUT | In Gruppe eintreten/ austreten (Nutzer) |  |  |
| /aktivitaet/klettern /gruppen/ID=1 | DELETE | Gruppe löschen (Admin, Nutzer-Pro, wenn erstellt) |  |  |
| /news | GET | Abrufen der Neuigkeiten |  |  |
| /news/gefahren | PUT | Gefahren aktualisieren, hinzufügen |  |  |
| /news/gefahren | GET | Gefahren abrufen |  |  |
| /news/wetterbericht | PUT | Wetterbericht aktualisieren |  |  |
| /news/wetterbericht | GET | Wetterbericht abrufen |  |  |
| /news/events | GET | Events abrufen |  |  |
| /news/events | PUT | Events aktualisieren |  |  |
| /unterkuenfte | GET | Unterkunftstypen abrufen |  |  |
| /unterkuenfte /camping | GET | Unterkunftsorte abrufen |  |  |
| /unterkuenfte /camping/ortX | GET | Informationen für Unterkunft abrufen |  |  |
| /unterkuenfte /camping/ortX | POST | Anfrageformular wird abgeschickt |  |  |
| /kontakt | GET | Kontaktmöglichkeiten einsehen, Anzeige von Notfallpunkten |  |  |
| /kontakt/ranger | POST | Mitteilung an Ranger senden |  |  |
| /kontakt/notfall | GET | Notfallnummer |  |  |

**Use case:** einer Gruppe beitreten

Actor: User

Precondition: muss im System angemeldet sein, mindestens eine Gruppe mit einem freien Platz muss vorhanden sein

Main flow:

1. User trägt sich in Gruppe ein
2. User erhält Bestätigung das Eintritt erfolgreich

Post condition: die Eintragung war erfolgreich

Exceptional flow:

1. A: User benötigt zu viel Zeit für die Anmeldung, maximale Gruppengröße ist dann schon erreicht. User erhält Benachrichtigung das Vorgang nicht erfolgreich war

Post condition: Eintragung war nicht erfolgreich

End: einer Gruppe beitreten

Use case: Gruppe filtern

Actor: User

Precondition: User muss angemeldet sein AND mind. eine Gruppe muss vorhanden sein, in der freie Plätze verfügbar sind

Main flow:

1. User wählt eine Aktivität aus
2. User kann diese filtern nach: Schwierigkeitsgrad, Datum, Dauer, verfügbare Plätze

Postcondition: Gruppe wurde gefiltert

Exceptional flow:

1.a: User verwendet keine weiteren Filter

Postcondition: alle Gruppen zu der Aktivität werden angezeigt

End: Gruppe filtern

**Use case:** Ranger kontaktieren

Actor: User, Ranger

Precondition: User muss angemeldet sein AND Ranger erreichbar

Main flow:

1. User füllt Formular aus (Pflichtfelder: Standort, Betreff (z.B. Notfall, Events))

Postcondition: Ranger wird kontaktiert, Formular ist versendet

Exceptional flow:

1. a: User füllt nicht alle Pflichtfelder aus

1. b: Internetverbindung bricht ab

Postcondition: Formular kann nicht abgeschickt werden

End: Ranger kontaktieren

**Use case:** Profi legt Gruppe an

Actor: User (Profi)

Precondition: User muss als Profi angemeldet sein

Main flow:

1. Profi erstellt Gruppe mit allen Pflichtangaben (Route, Uhrzeit, Treffpunkt, maximale Gruppengröße, Dauer)
2. Admin überprüft Daten
3. Admin gibt Gruppe frei

Postcondition: Gruppe wurde erfolgreich erstellt

Exceptional flow:

1. a: Profi füllt nicht alle Pflichtangaben aus
2. a: Admin stellt fehlerhafte Eingabe fest (Semantik)
3. b: Admin stellt Termin- oder Routenkollision fest

Postcondition: Gruppe wurde nicht freigegeben

End: Profi legt Gruppe an