

Theoretische Informatik – ti2023 22

Übungsblatt 01

Aufgabe 1:

Wir haben uns den Standard RFC 7807 ausgesucht. Dieser Standard definiert die Struktur und den Aufbau von Fehler- oder Problemnachrichten in http-basierten Web-APIs (Application Programming Interface). Hauptzweck dieses Standards ist es, eine einheitliche Konvention für die Fehlerberichterstattung und -behandlung in http-Anfragen und -Antworten zu etablieren.

Ein Beispiel für diesen Standard würde so aussehen:

HTTP/1.1 403 Forbidden

Content-Type: application/problem+json

Content-Language: en

```
{
  "type": "https://example.com/probs/out-of-credit",
  "title": "You do not have enough credit.",
  "detail": "Your current balance is 30, but that costs 50.",
  "instance": "/account/12345/msgs/abc",
  "balance": 30,
  "accounts": ["/account/12345",
               "/account/67890"]
}
```

a) Geben wir die BNF-Regeln wieder, indem wir in „echte“-BNF konvertieren

- 1) **<problem+json> ::= "{" <type> "," <title> "," <detail> "," <instance> "}"**
- 2) **<type> ::= <string> ":" <uri> <string>**
- 3) **<string> ::= <char> | "." | "(" | ")" | "[" | "]" | "{" | "}" | "|" | "!" | "!" | "/" | "\" | "?" | "@" | ":" | "\"" | <string>**
- 4) **<char> ::= A|B|C|...|Z|a|b|c|...|z|0|1|2|3|...|9| "-" | "_" | <char>**
- 5) **<uri> ::= <http> <char> "." <tld> <string> <char>**
- 6) **<http> ::= https://**
- 7) **<tld> ::= A|B|C|...|Z|a|b|c|...|z|0|1|2|3|...|9|<tld>**
- 8) **<title> ::= <detail>**
- 9) **<detail> ::= <instance>**
- 10) **<instance> ::= <string> ":" <string>**

b) Geben wir die Variablen an

Es ist $G = (\Sigma, V, P, \text{problem+json})$ Dabei betrachten wir erstmal die Menge V , die alle unserer Variablen enthält.

$V = \{ \text{problem+json} ; \text{type} ; \text{title} ; \text{detail} ; \text{instance} ; \text{string} ; \text{uri} ; \text{char} ; \text{tld} ; \text{http} \}$

c) Geben wir nun die Buchstaben an

Σ bildet unser Alphabet

$\Sigma = \{A; B; C; D; E; F; G; H; I; J; K; L; M; N; O; P; Q; R; S; T; U; V; W; X; Y; Z; a; b; c; d; e; f; g; h; i; j; k; l; m; n; o; p; q; r; s; t; u; v; w; x; y; z; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; ", " ; "- " ; "_ " ; "https://" ; "." ; "(" ; ")" ; "[" ; "]" ; "{" ; "}" ; " " ; "!" ; "/" ; "\" ; "?" ; "@" ; ":" ; "\" \}$

d) Geben wir das Startsymbol an

Unser Startsymbol ist hier **problem+json**

Geben wir vier fertige Beispiele an, die wir mit den Regeln konstruiert haben.

Anmerkung: in unseren Beispielen kommen die Namen der Team-Mitglieder vor.

Beispiel1:

```
<problem+json> R1 →{ <type> ", " <title> ", " <detail> ", " <instance> } R2 →
{"type" : "https://beispiel.com/Junior" , <title> , <detail> , <instance> }

R8 →{ "type" : "https://beispiel.com/Junior" , "title" : "You should not
pass Ekane." , <detail> , <instance> } R9 → { "type" :
"https://beispiel.com/Junior" , "title" : "You should not pass Ekane." ,
"detail" : "Lesage don't give you the permssion to acces this file." ,

<instance> } R10 →{ "type" : "https://beispiel.com/Junior" , "title" :
"You should not pass Ekane." , "detail" : "Lesage don't give you the
permssion to acces this file." "instance" ":" "/account/123/prompt/Njoh" }
```

Beispiel2:

```
<problem+json> R1 →{ <type> , <title> , <detail> , <instance> } R2 →

{"type" : "https://hp.com/Steve" , <title> , <detail> , <instance> } R8 →{
"type" : "https://hp.com/Steve" , "title" : "Aguiwo II." , <detail> ,

<instance> } R9 → { "type" : "https://hp.com/Steve" , "title" : "Aguiwo
II." , "detail" : "Ekane Njoh ist nicht eingetragen." , <instance> } R10
→{ "type" : "https://hp.com/Steve" , "title" : "Aguiwo II." , "detail"
: "Ekane Njoh ist nicht eingetragen." , "instance" ":"
"/account/Lesage/mgsa/Njoh" }
```

Beispiel3:

```
<problem+json> R1 →{ <type> , <title> , <detail> , <instance> } R2 →
{"type" : "https://Steve.123/Aguiwo" , <title> , <detail> , <instance> } R8
→{ "type" : "https://Steve.123/Aguiwo" , "title" : "Junior hat bald
```

```
Geburtstag." , <detail> , <instance> } R9 → { "type" :
"https://Steve.123/Aguiwo" , "title" : "Junior hat bald Geburtstag." ,
"detail" : "TI macht Spaß." , <instance> } R10 →{ "type" :
"https://Steve.123/Aguiwo" , "title" : "Junior hat bald Geburtstag." ,
"detail" : "TI macht Spaß." , "instance" ":" "/Lesage/1234/localhost/moin"
}
```

Beispiel4:

```
<problem+json> R1 →{ <type> , <title> , <detail> , <instance> } R2 →
{"type" : "https://lib.is2/Njoh" , <title> , <detail> , <instance> } R8 →{
"type" : "https://lib.iso/Njoh" , "title" : "failed to call Steve." ,
<detail> , <instance> } R9 → { "type" : "https://lib.iso/Njoh" ,
"title" : "failed to call Steve." , "detail" : "can not reach Aguiwo." ,
<instance> } R10 →{ "type" : "https://lib.iso/Njoh" , "title" : "failed
to call Steve." , "detail" : "can not reach Aguiwo." , "instance" ":"
"/log/error/Steve/9875" }
```

Literaturverzeichnis

<https://www.rfc-editor.org/rfc/rfc7807>

[Application error: a client-side exception has occurred \(codecentric.de\)](#)

[JSON - GeeksforGeeks](#)