



Rust in the Web

Eric Kunze



- 1 Entwicklungsgeschichte
- 2 Warum Rust?
- 3 Was macht Rust sicher?
- 4 Rust in the Web
- 5 Iron
- 6 Demo



- 2006 persönliches Projekt von Graydon Hoare
- ab 2009 Projekt bei Mozilla
- 15. Mai 2015 Veröffentlichung der Version 1.0
- Entwicklung einer neuen Browserenging → Servo



- Warum nicht C++ oder Java?
- Rust
 - schnell
 - Kontrolle über das System
 - minmale Runtime
 - keine Garbage Collection
 - sicher
 - Features von höheren- und funktionalen Programmiersprachen



■ Ownership and Borrowing

■ Ownership

- jede Ressource hat genau einen Besitzer

```
1 fn foo() {  
2     let mut y: Vec<i32>  
3     = Vec::new();  
4  
5     y.push(4);  
6     bar(y);  
7     y.push(5); // Compiler Error  
8 }
```

```
1 fn bar(x: Vec<i32>) {  
2     ...  
3 }
```



- Borrowing
 - jede Ressource kann verliehen werden
- shared borrow

```
1 fn foo() {  
2     let mut y: Vec<i32>  
3     = Vec::new();  
4  
5     y.push(4);  
6     bar(&y);  
7     y.push(5);  
8 }
```

```
1 fn bar(x: &Vec<i32>){  
2     println!("{}",x[0]); // Ok  
3     x.push(1); // Compiler Error  
4 }
```



■ mutable borrow

- nur eine Referenz auf eine Ressource und diese hat nur einen Besitzer

```
1 fn foo() {  
2     let mut y  
3     = vec![1, 2, 3, 4, 5];  
4     let mut x: Vec<i32>  
5     = Vec::new();  
6  
7     bar(&y, &mut x); // Ok  
8     bar(&y, &mut y);  
9     // Compiler Error  
10 }
```

```
1 fn bar(y: &Vec<i32>,  
2       x: &mut Vec<i32>){  
3  
4     for v in y{  
5         x.push(*v);  
6     }  
7 }
```



- Aliasing und Mutation zur gleichen Zeit wird verhindert
- Bsp. C++

```
1 void foo(){
2     int *y = new int[10];
3     for(int i=0;i<10;i++){
4         y[i] = i;
5
6         int *x = &y[9];
7         y=bar(y);
8         delete[] y;
9         cout<<*x<<endl;
10 }
```

```
1 int* bar(int *v){
2     delete[] v;
3     v = new int[5]
4     for(int i=0;i<10;i++){
5         v[i] = i*2;
6     }
7     return v;
8 }
```




■ HTTP-Server

- Hyper
- tiny-http

■ HTTP client

- Hyper
- curl-rust

■ Database drivers

- rust-postgres
- redis-rs

■ Frameworks

- Iron
- Conduit
- rustful
- Nickel



- Webframework
- basiert auf Hyper
- hochgradig Nebenläufig
- keine unnötigen Features im Basisframework, aber leicht erweiterbar
- bietet Infrastruktur, um das Framework an individuelle Bedürfnisse anzupassen

