

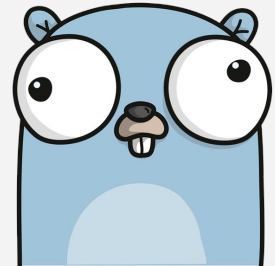
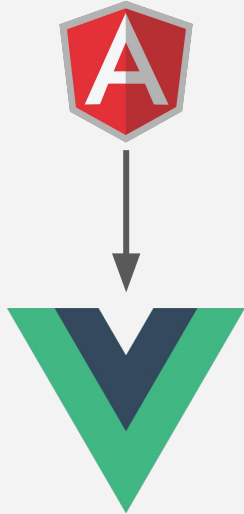
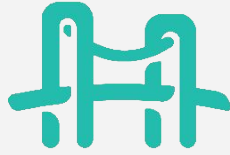
USANDO O MESMO  
CÓDIGO NO FRONT E  
BACKEND

---

JS

# AGENDA

- Contexto da Transfeera
- Problema encontrado
- Webpack, NPM e Node
- Helpers
- Validações complexas
- Traduções
- Problema de múltiplas dependências
- Conclusão



# PROBLEMA

- Réplica de código entre front e backend
  - Dificuldade de previsibilidade
  - Maior trabalho para fazer alterações
  - Possível despadronização

E se compartilhássemos código  
entre o front e o backend?



Helpers

```
const cnpjPattern = new StringMask('00.000.000/0000-00');  
const cpfPattern = new StringMask('000.000.000-00');
```

```
class PersonTypeService {
```

```
    static format(document) {
```

```
    }
```

```
    static getType(document) {
```





```
const regexes = {  
  bankAccount: {  
    AGENCY_ONE:  
    DIGIT:  
    NUMERIC_DIGIT:  
    AGENCY:  
    ACCOUNT:  
  },  
  FULL_NAME:  
  EMAIL:  
  
  DATE:  
  
  ONLY_NUMBERS_BUT_ALL_ZEROS:  
  CPF_CNPJ:  
  
};  
  
module.exports = regexes;
```



**MAS** Cuidado com as  
funcionalidades específicas do  
front e do back

Validações bancárias complexas

- Validação de esquema bancário no front e no backend
- Validação cruzando campos imputados no front e backend
  - Cálculos de dígitos no front e no backend

## Quais os dados bancários do favorecido?

Qual o banco do favorecido?

Qual a agência?

Dígito?

Qual o tipo da sua conta?

Qual a conta ?

Dígito?

## Quanto você quer transferir?

Qual o valor? (máximo de R\$ 25.000,00)

## Qual o ID de integração?

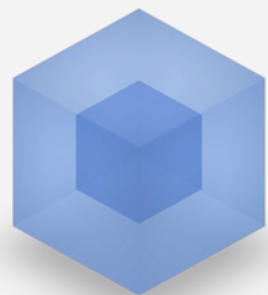
ID de integração  
(opcional)

```
const defaultSchema = {
  agency: {
    maxLength:
    required:
    pattern:
    digit: {
      exactLength:
      required:
      pattern:
    }
  },
  account: {
    maxLength:
    required:
    pattern:
    digit: {
      exactLength:
      required:
      pattern:
    }
  },
  accountType: {
    required:
    allowedTypes:
  }
};
```

```
module.exports = defaultSchema;
```



```
<tr-text-field
  label="Dígito"
  name="agency_digit"
  ref="agency_digit"
  :error="!complexValidations.agency_digit.isValid"
  :error-to-force-until-is-dirty="errorField === 'agency_digit'"
  :classes="fullSize ? 'col-lg-2' : 'col-lg-1'"
  :required="validations.agency.digit.required"
  :pattern="validations.agency.digit.pattern"
  :error-message="complexValidations.agency_digit.errorMessage ||
validations.agency.digit.errorMessage"
  :maxlength="validations.agency.digit.maxLength ||
validations.agency.digit.exactLength"
  :on-value-change="validate.bind(this, 'agency_digit')"
  v-model="bankAccount.agency_digit" />
```



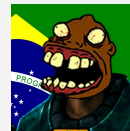


# IDIOMAS

- Front 
- Api do produto principal 
- Micro serviços  

# TRADUÇÕES

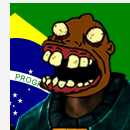
API retorna ERRO**R**  front mostra ERRO



```
{  
  code: "ERR_666"  
  message: "WTF"  
}
```



```
{  
  code: "ERR_666"  
  message: "Que  
    po*** é essa?"  
}
```



erros/idioma/contexto.json {

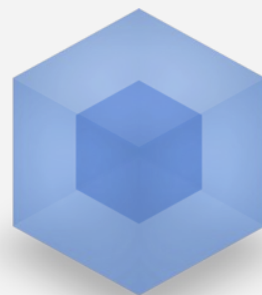
```
{
  WTF_ERROR: {
    code: "ERR_666"
    message: "WTF"
  },
  IT_WORKS_ON_MY_MACHIN
E_ERROR: {
  code: "ERR_434",
  message: "it works on
my machine"
}
}
```

```
WTF_ERROR: {
  code: "ERR_666"
  message: "Que por*** é
essa"
},
IT_WORKS_ON_MY_MACHIN
E_ERROR: {
  code: "ERR_434",
  message: "Na minha
máquina funciona"
}
}
```



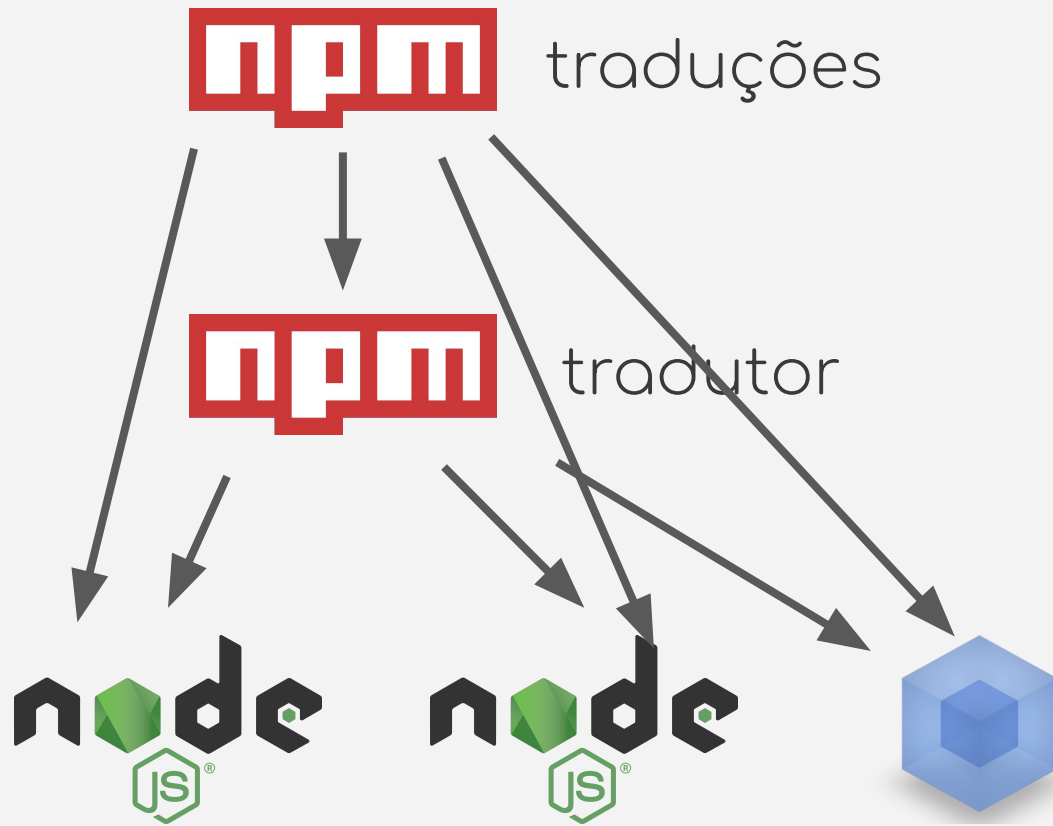
```
const translator = new Translator(languages.english);  
translator.translate(error.code);
```





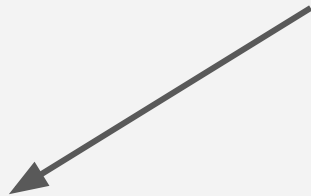
FICOU BOM?



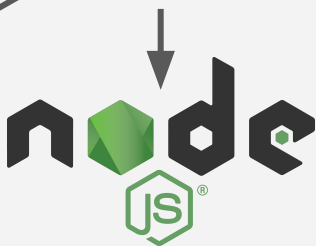
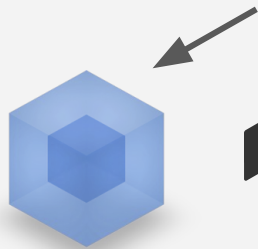


ENTRE DIFERENTES TECNOLOGIAS

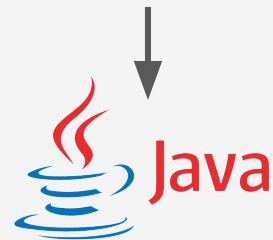
traduções JSONs



tradutor



projeto



ESTAMOS REINVENTANDO A RODA PRA  
ESSE PROBLEMA DE TRADUÇÃO?

# CONCLUSÃO





fim