



Internationalization in React

plugins and loaders for react-intl and the babel webpack workflow



DDI Ognian Tschakalov

CEO and Founder OGI-IT

OGI-IT is an independent software vendor (ISV) and system integrator (SI)
first product - **LIMS** (Laboratory Information Management System) based
on the OGI-IT IMS Framework

<http://www.ogi-it.com>

ognian.tschakalov@ogi-it.com

@ogi_it





why i18n ?

ogi (all) Studien

1 - 10 / 10

<input type="checkbox"/>	Name	Beschreibung	Typ	Zustand
<input type="checkbox"/>	Just_a_Test_Study	Hallo	Val GLP	committed
<input type="checkbox"/>			Val GLP	final
<input type="checkbox"/>		st2 B	Val GMP	final
<input type="checkbox"/>		exalen in pipi	short	
<input type="checkbox"/>			Study GLP	
<input type="checkbox"/>		enproben zur Studie mit Namen "xxx" passen		
<input type="checkbox"/>		signec	Study GLP	
<input type="checkbox"/>			Val GMP	audited
<input type="checkbox"/>				signed
<input type="checkbox"/>				signed

Aktionen auf ein einzelnes Element

- Bearbeiten ...
- Löschen ...
- Unterschreiben ...

Aktionen für mehrere Elemente

- Mehrere Objekte als JSON exportieren ...

+ Add

+ Hinzufügen



Internationalize React apps. This library provides React components and an API to format dates, numbers, and strings, including pluralization and handling translations.

423 commits

2 branches

Features

- Display numbers with separators.
- Display dates and times correctly.
- Display dates relative to "now".
- Pluralize labels in strings.
- Support for 150+ languages.
- Runs in the browser and Node.js.
- Built on standards.

Example

There are several [runnable examples](#) in this Git repo, but here's a Hello World one:

```
import React, {Component} from 'react';
import ReactDOM from 'react-dom';
import {IntlProvider, FormattedNumber, FormattedPlural} from 'react-intl';

class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name      : 'Eric',
      unreadCount: 1000,
    };
  }

  render() {
    const {name, unreadCount} = this.state;

    return (
      <p>
        Hello <b>{name}</b>, you have { ' ' }
        <FormattedNumber value={unreadCount} /> { ' ' }
        <FormattedPlural value={unreadCount}
          one="message"
          other="messages"
        />.
      </p>
    );
  }
}

ReactDOM.render(
  <IntlProvider locale="en">
    <App />
  </IntlProvider>,
  document.getElementById('container')
);
```

But I want a simple text in ,en' and ,de'



Is there another framework?



Yes, and all of them are build
from a developer perspective ...



but professional translators also use
computers AND computer tools to
make translations better...

CAT (Computer Aided Translation) Tools



make it developer and translator
friendly

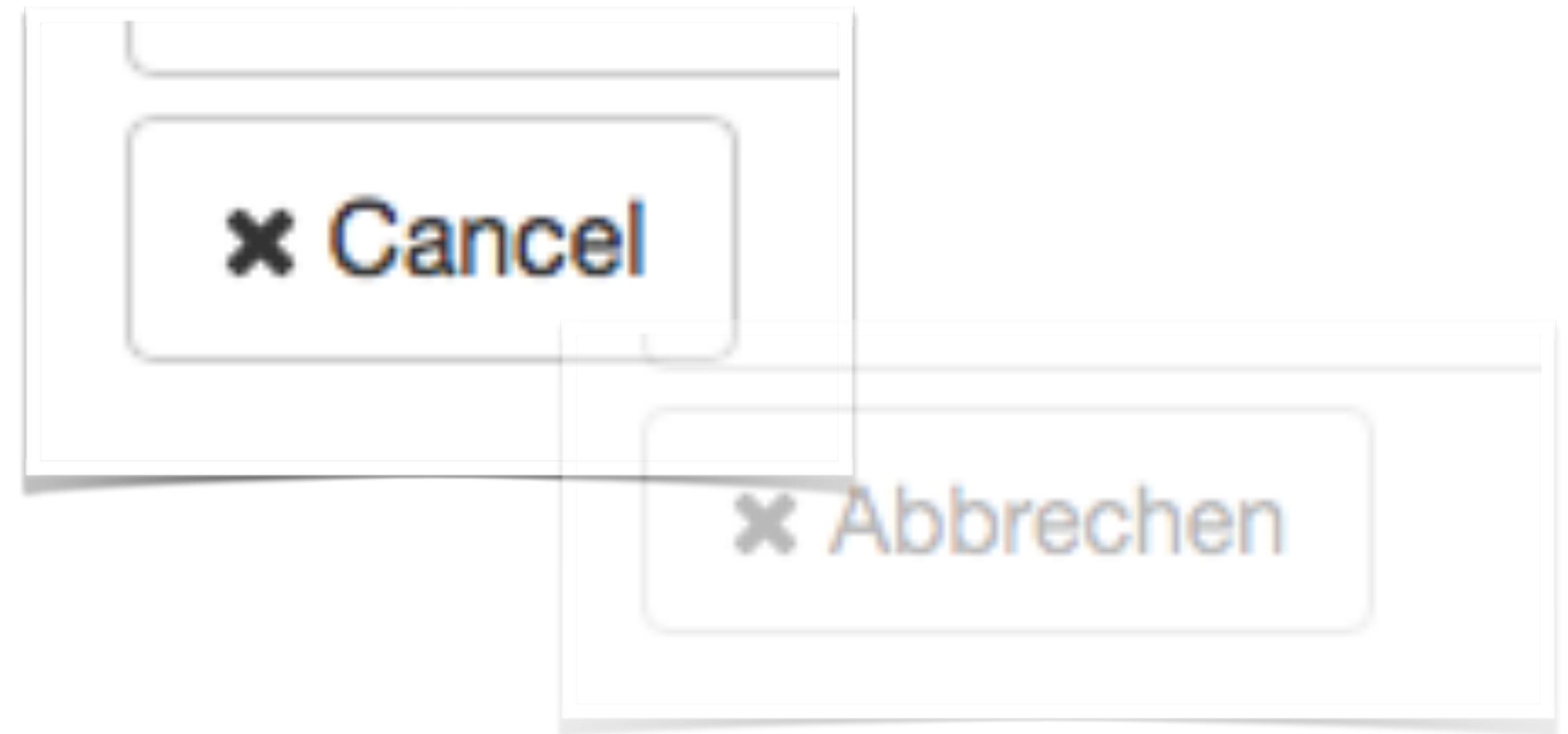


workflow 1/6
writing source code

```
import {FormattedMessage} from 'react-intl';
```

```
...
```

```
<Button onClick={this.goBack}>  
  <i className="fa fa-times"></i>  
  &nbsp;  
  <FormattedMessage  
    id="cancelInDetail"  
    defaultMessage="Cancel"  
  />  
</Button>
```



1. Use `<FormattedMessage id="newItem" defaultMessage="Add item" />` for adding text in your react components.



workflow 2/6 babel-plugin-react-intl

webpack.dev.config.js:

```
module: {  
  loaders: [  
    ...  
    {  
      test: /\.js?$/, exclude: /node_modules\/(?!rfc6902)/, loader: 'babel-loader',  
      query: {  
        "plugins": [  
          ["react-intl", {  
            // "messagesDir": "dist/intl/extractedMessages",  
            "enforceDescriptions": false  
          }]  
        ],  
        "presets": ['react', "es2015", "stage-1"]  
      }  
    }  
  ]  
},  
},
```

2. Use the babel plugin `babel-plugin-react-intl` to extract them from your source code.



workflow 3/6

Ognian/react-intl-webpack-plugin

webpack.dev.config.js:

```
var ReactIntlPlugin=require('react-intl-webpack-plugin');
...
plugins: [
...
  new ReactIntlPlugin()
],
...
module: {
  loaders: [
    ...
    {
      test: /\.js?$/, exclude: /node_modules\/(?!rfc6902)/, loader: 'babel-loader',
      query: {
        "cacheDirectory": true,
        "metadataSubscribers": [ReactIntlPlugin.metadataContextFunctionName],
        "plugins": [
          "transform-runtime",
          ["react-intl", {
            "enforceDescriptions": false
          }]
        ],
        "presets": ['react', "es2015", "stage-1"]
      }
    }
  ]
}
...
],
},
```

3. Use `react-intl-webpack-plugin` to combine them into one message file called `reactIntlMessages.json` and put this file into the webpack output path.



reactIntlMessages.json

```
[
  {
    "id": "cancelInHelp",
    "defaultMessage": "Cancel"
  },
  {
    "id": "cancelInDetail",
    "defaultMessage": "Cancel"
  },
  ::
  {
    "id": "newItem",
    "description": "the text on the plus button for adding an item",
    "defaultMessage": "Add"
  },
  ::
  {
    "id": "noActionsHeader",
    "defaultMessage": "no actions for {itemCount, plural, one {this item} other {those items}}"
  }
]
```



reactIntlMessages-en-de-C.json

```
[
  {
    "id": "cancelInHelp",
    "defaultMessage": "Abbrechen"
  },
  {
    "id": "cancelInDetail",
    "defaultMessage": "Abbrechen"
  },
  ::
  {
    "id": "newItem",
    "description": "the text on the plus button for adding an item",
    "defaultMessage": "Hinzufügen"
  },
  ::
  {
    "id": "noActionsHeader",
    "defaultMessage": "Keine Aktionen für {itemCount, plural, one{dieses Element} other{diese Elemente}}."
  }
]
```



reactIntlMessage.json -> reactIntlMessages-en-de-C.json

use memsource.com to translate

The screenshot shows the Memsource.com interface. At the top, project details are displayed: # 2, Ausgangssprache EN, Name Projekt, Zielsprachen DE, Erstellt von Ognian, Maschinelle Übersetzung Microsoft with Feedback, Erstellt 31 Mai 14:08, Eigentümer Ognian, Status Neu. Below this is a 'Jobs' table with columns for #, Bestätigt, Datei, and Status. A job is listed with # 14, 100% completion, and file name reactIntlMessages.json. A 'Tools' dropdown menu is open, showing options for 'JSON' configuration: 'Use HTML subfilter' (unchecked), 'Convert to Memsource tags' (empty input), 'Import specific keys only (use regexp)' (filled with */defaultMessage), 'Context note' (filled with ../description), and 'Max. target length' (empty input).

4. Use CAT (Computer Aided Translation) tools (like the cloud based memsource.com) to translate this file into the translated file. These tools use a concept called Translation Memory (TM) . This is a separate file where all translations are stored and with this file all translations can be reapplied to a newly generated `reactIntlMessages.json` file.



cloud.memsource.com/web/project2/show/1108156

MEMSOURCE CLOUD

PROJEKTE TRANSLATION MEMORIES TERMBANKEN

Setup Benutzer Abmelden Ognian
Schließen Sie jetzt ein Abonnement ab.

NEU ERSTELLEN +

Projekt

Bearbeiten Speichern unter Löschen

SUCHE

AKTUELLE ELEMENTE

- Projekt
- LIMsv2_de_TM

NÜTZLICHE LINKS

- Jobbörse (Beta) **Neu**
- Nächste Wartung
- Memsource Editor testen
- Erste Schritte
- Bedienungsanleitung
- Support

Papierkorb

Last login: 17-06-2016 17:31 [Details](#)

2 Ausgangssprache EN
Name Projekt Zielsprachen DE
Erstellt von Ognian Maschinelle Übersetzung [Microsoft with Feedback](#)
Erstellt 31 Mai 14:08 Eigentümer Ognian
Status Neu

Dashboard

Jobs **Neu** Bearbeiten Herunterladen Analysieren Vorübersetzen **Tools** Löschen

<input checked="" type="checkbox"/>	#	Bestätigt	Datei	Status	Zielsprache	Linguist	Fällig
<input type="checkbox"/>	14	100%	<code>reactIntlMessages.json</code>	Neu	DE		

Analysen

Translation Memories **Auswählen** **Neu erstellen**

Name	Lesen	Schreiben	Abzug (%)
<code>LIMsv2_de_TM</code>	✓	✓	

Termbanken **Auswählen** **Neu erstellen**

Referenzen **Neu**



```
[
  {
    "id": "cancelInHelp",
    "defaultMessage": "Cancel"
  },
  {
    "id": "cancelInDetail",
    "defaultMessage": "Cancel"
  },
  ::
  {
    "id": "newItem",
    "description": "the text on the plus button for
adding an item",
    "defaultMessage": "Add"
  },
  ::
  {
    "id": "noActionsHeader",
    "defaultMessage": "no actions for {itemCount,
plural, one {this item} other {those items}}"
  }
]
```

```
[
  {
    "id": "cancelInHelp",
    "defaultMessage": "Abbrechen"
  },
  {
    "id": "cancelInDetail",
    "defaultMessage": "Abbrechen"
  },
  ::
  {
    "id": "newItem",
    "description": "the text on the plus button for
adding an item",
    "defaultMessage": "Hinzufügen"
  },
  ::
  {
    "id": "noActionsHeader",
    "defaultMessage": "Keine Aktionen für {itemCount,
plural, one{dieses Element} other{diese Elemente}}."
  }
]
```




workflow 5/6
Ognian/react-intl-webpack-plugin



5. Save the TM file and the translated json file in a separate directory of your source code.



```
[
  {
    "id": "cancelInHelp",
    "defaultMessage": "Abbrechen"
  },
  {
    "id": "cancelInDetail",
    "defaultMessage": "Abbrechen"
  },
  ...
  {
    "id": "newItem",
    "description": "the text on the plus button for
adding an item",
    "defaultMessage": "Hinzufügen"
  },
  ...
  {
    "id": "noActionsHeader",
    "defaultMessage": "Keine Aktionen für {itemCount,
plural, one{dieses Element} other{diese Elemente}}."
  }
]
```

reactIntlJson-loader

```
{
  "cancelInHelp": "Abbrechen",
  "cancelInDetail": "Abbrechen",
  ...
  "newItem": "Hinzufügen",
  ...
  "noActionsHeader": "Keine Aktionen für
{itemCount, plural, one{dieses Element}
other{diese Elemente}}."
}
```



workflow 6/6 Ognian / reactIntlJson-loader

```
import {IntlProvider, addLocaleData} from 'react-intl';
import deLocaleData from 'react-intl/locale-data/de';
import deMessages from 'reactIntlJson!./intl/reactIntlMessages-en-de-C.json';

let messages = {};
messages.de = deMessages;
const currentLang = navigator.language;
console.log("currentLang: ", currentLang);

addLocaleData(deLocaleData);

...

ReactDOM.render(
  <Provider store={store}>
    <IntlProvider locale={currentLang} messages={messages[currentLang]}>
      <div>
        <App history={history}/>
      </div>
    </IntlProvider>
  </Provider>,
  document.getElementById('root')
)
```

6. Use `reactIntlJson-loader` to load the translated files and convert them to messages.



Browser polyfill for Intl

```
if (!global.Intl) {  
  console.log("intl plugin needed");  
  require('intl');  
  require('intl/locale-data/jsonp/en.js');  
  require('intl/locale-data/jsonp/de.js');  
}
```



ogi (all) Studien

1 - 10 / 10

<input type="checkbox"/>	Name	Beschreibung	Typ	Zustand
<input type="checkbox"/>	Just_a_Test_Study	Hallo	Val GLP	committed
<input type="checkbox"/>			Val GLP	final
<input type="checkbox"/>		st2 B	Val GMP	final
<input type="checkbox"/>		exalen in pipi	short	
<input type="checkbox"/>			Study GLP	
<input type="checkbox"/>		enproben zur Studie mit Namen "xxx" passen		
<input type="checkbox"/>		signec	Study GLP	
<input type="checkbox"/>			Val GMP	audited
<input type="checkbox"/>				signed
<input type="checkbox"/>				signed

Aktionen auf ein einzelnes Element

- Bearbeiten ...
- Löschen ...
- Unterschreiben ...

Aktionen für mehrere Elemente

- Mehrere Objekte als JSON exportieren ...

+ Add

+ Hinzufügen



Thank you!

<http://www.ogi-it.com>

ognian.tschakalov@ogi-it.com

@ogi_it