



# REACT MINIPROJEKT-1

GRUPPMEDLEMMAR:

JOOOONAS GLINGERT

LIIIIISA PÄÄJÄRVI

TOOOONY MARTINSSON

# IDÉ: EN BOKTIPSAPP BASERAD PÅ BOKTIPSETS API

## Help

To get help for a certain call, call the script without parameters.

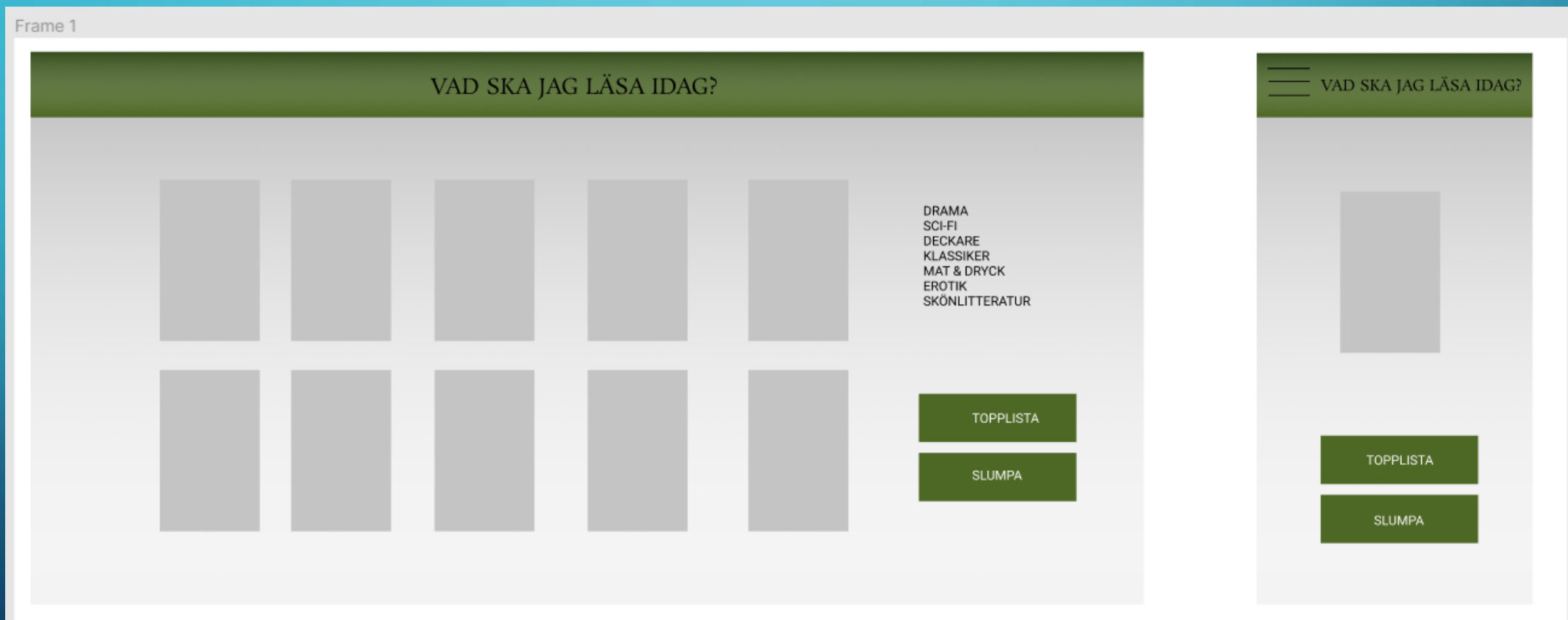
## Available functions

- **Authenticate:** <http://api.boktipset.se/api/authenticate.cgi>
- **Information about a book:** <http://api.boktipset.se/book/book.cgi>
- **All comments for a book:** <http://api.boktipset.se/book/comments.cgi>
- **Show who is tracking a book:** <http://api.boktipset.se/book/tracking.cgi>
- **Search for books:** <http://api.boktipset.se/book/search.cgi>
- **Set a grade:** [http://api.boktipset.se/book/set\\_grade.cgi](http://api.boktipset.se/book/set_grade.cgi)
- **Post a comment:** [http://api.boktipset.se/book/submit\\_comment.cgi](http://api.boktipset.se/book/submit_comment.cgi)
- **Events for a book:** <http://api.boktipset.se/book/events.cgi>
- **Lists that a book is part of:** <http://api.boktipset.se/book/lists.cgi>
- **Information about a author:** <http://api.boktipset.se/author/author.cgi>
- **Information about a member:** <http://api.boktipset.se/member/profile.cgi>
- **Updating the information about a member:** [http://api.boktipset.se/member/save\\_profile.cgi](http://api.boktipset.se/member/save_profile.cgi)
- **Following, or un-following a member:** [http://api.boktipset.se/member/save\\_follow.cgi](http://api.boktipset.se/member/save_follow.cgi)
- **Getting all the followers of a member, or all the members another member is following.:** <http://api.boktipset.se/member/follow.cgi>
- **A members shelf:** <http://api.boktipset.se/member/shelf.cgi>
- **Events for a member:** <http://api.boktipset.se/member/events.cgi>
- **All events for grouped events:** [http://api.boktipset.se/events/get\\_all\\_events.cgi](http://api.boktipset.se/events/get_all_events.cgi)
- **Search for members:** <http://api.boktipset.se/member/search.cgi>
- **All lists a member has made:** <http://api.boktipset.se/member/lists.cgi>
- **All books that a members friends are reading:** [http://api.boktipset.se/member/friends\\_reading.cgi](http://api.boktipset.se/member/friends_reading.cgi)
- **A list made by a member:** <http://api.boktipset.se/lists/memberlist.cgi>
- **A list based on statistics:** <http://api.boktipset.se/lists/statlist.cgi>
- **Create or edit lists:** [http://api.boktipset.se/lists/save\\_list.cgi](http://api.boktipset.se/lists/save_list.cgi)
- **Show a list of all genres available:** <http://api.boktipset.se/lists/genres.cgi>
- **Show a list of books for a genre:** <http://api.boktipset.se/lists/genre.cgi>
- **Add or remove from tracking lists:** [http://api.boktipset.se/lists/save\\_tracking.cgi](http://api.boktipset.se/lists/save_tracking.cgi)
- **Get information about a place:** <http://api.boktipset.se/places/place.cgi>
- **Get places for a book:** [http://api.boktipset.se/places/places\\_for\\_book.cgi](http://api.boktipset.se/places/places_for_book.cgi)
- **Get places for a author:** [http://api.boktipset.se/places/places\\_for\\_author.cgi](http://api.boktipset.se/places/places_for_author.cgi)
- **List check ins for a place, member, book, or author:** [http://api.boktipset.se/places/list\\_check\\_ins.cgi](http://api.boktipset.se/places/list_check_ins.cgi)
- **Get places for a bounding box:** [http://api.boktipset.se/places/places\\_for\\_coords.cgi](http://api.boktipset.se/places/places_for_coords.cgi)
- **Add a new place:** [http://api.boktipset.se/places/add\\_place.cgi](http://api.boktipset.se/places/add_place.cgi)
- **Report an error on a place:** [http://api.boktipset.se/places/report\\_place.cgi](http://api.boktipset.se/places/report_place.cgi)
- **Get blogposts:** [http://api.boktipset.se/blogapi/get\\_posts.cgi](http://api.boktipset.se/blogapi/get_posts.cgi)
- **Get blogpost:** [http://api.boktipset.se/blogapi/get\\_post.cgi](http://api.boktipset.se/blogapi/get_post.cgi)
- **Add highlight:** [http://api.boktipset.se/highlights/add\\_highlight.cgi](http://api.boktipset.se/highlights/add_highlight.cgi)
- **Get highlights for a book:** [http://api.boktipset.se/highlights/highlights\\_for\\_book.cgi](http://api.boktipset.se/highlights/highlights_for_book.cgi)
- **Get highlights for a member:** [http://api.boktipset.se/highlights/highlights\\_for\\_member.cgi](http://api.boktipset.se/highlights/highlights_for_member.cgi)

## Example

This is a sample call to the API. This call will always return this help-page.

# STEG 1: DESIGNSKISS FÖR BOKTIPSAPPEN




# STEG 2: GRUNDLAYOUT OCH KOMPONENTER

- App.tsx
  - Layout
    - Header
    - Main
      - Content
        - BookView
          - SectionItem
        - BookDetail
      - Side

# STEG 3: KAOS OCH NY INRIKTNING


- Oklart om API:et fungerar
- Ny inriktning
- Dikter
- Helt öppet, ingen accesskey



The screenshot shows the Poemist API documentation page. A black box with the text "Poemist API" is overlaid on the top left. The page has a red sidebar with navigation links: "Search", "Introduction", "Misc. Services", and "Errors". The main content area includes an "Introduction" section, a "Misc. Services" section, and a "Get Random Poems" section. The "Get Random Poems" section shows an "HTTP Request" and "Query Parameters". A terminal window on the right shows a curl command and its JSON response.

This web public API was created by Poemist. The Poemist API endpoint is located at <https://www.poemist.com/api/v1/>. You can find the Poemist portal / homepage [here](#). If you need Poemist API support, you can reach out to their Twitter account at [@poemistcom](#). The Poemist API requires API Key authentication. For more information, check out their [API Documentation](#). The

# STEG 4: DAGENS DIKT



Dagens dikt


أحلا - إلى الفارس الجبان - Poem by أحلام مستغانمي

Oh! Heed Not Those Beams  
Edward Henry Bickersteth

Cook  
Cicely Fox Smith

My Beginning  
Alvie C. Hutton

Cedar umbrellas  
Matsuo Basho



Ändra tema

Slumpa

Marsvin



# KOMPONENTER - API

```
25 class Content extends Component<Props, State> {
26   state: State = {
27     poems: [],
28     theme: this.props.theme
29   }
30
31   async fetchPoems() {
32     try {
33       const response = await fetch('https://www.poemist.com/api/v1/randompoems')
34     };
35
36     const result = await response.json();
37     this.setState({ poems: result })
38
39   } catch (error: unknown) {
40     console.error(error);
41   }
42 }
43
44 componentDidMount() {
45   this.fetchPoems();
46 }
47
48 render() {
49   return (
50     <div style={ contentStyle(this.props) }>
51       <Route exact path="/">
52         <PoemView poems={this.state.poems}/>
53       </Route>
54       <Route path="/marsvin" component={Marsvin}/>
55     </div>
56   )
57 }
58
59 }
60
```

You, 8 days ago • flexigt

# KOMPONENTER - DIKTÖVERSIKTEN

```
<> index.html TS App.tsx TS Layout.tsx TS Header.tsx TS Main.tsx TS Content.tsx
src > components > TS PoemView.tsx > [🔍] default
  Lisa Pääjärvi, 31 minutes ago | 4 authors (You and others)
1  import React, { CSSProperties } from 'react';
2  import { Poem } from './Content';
3  import Sectionitem from './Sectionitem';
4  import ErrorBoundary from '../ErrorBoundary';
5
  You, 2 days ago | 1 author (You)
6  interface Props {
7    |   poems: Poem[];
8  }
9
10 function PoemView(props: Props) {
11   return (
12     <div style={ viewStyle }>
13       <ErrorBoundary>
14         {props.poems.map(poem => <Sectionitem poem={poem} key={poem.title}/>)}
15       </ErrorBoundary>
16     </div>
17   )
18 }
19
```



# KOMPONENTER – POEM MODAL

```
class Sectionitem extends Component<Props, State> {

  state: State = {
    isModalVisible: false
  }

  openModal = () => {
    this.setState({ isModalVisible: true });
  }

  closeModal = (event: React.MouseEvent) => {
    event.stopPropagation();
    this.setState({ isModalVisible: false });
  };

  render() {
    return (
      <div style={ itemStyle } onClick={ this.openModal }>
        <p>{this.props.poem.title}</p>
        <p style= { poemStyle }>{this.props.poem.poet.name}</p>

        {this.state.isModalVisible && (
          <ErrorBoundary>
            <Modal shouldClose={this.closeModal}>
              <b><p style= { poemStyle }>{this.props.poem.title}</p></b>
              <p style= { poemStyle }>{this.props.poem.content}</p>
              <p style= { poemStyle }>{this.props.poem.poet.name}</p>
              <button onClick={this.closeModal}>STĂNG</button>
            </Modal>
          </ErrorBoundary>
        )}
      </div>
    )
  }
}
```

# KOMPONENTER - SIDE

```
changeTheme = () => {
  if (this.state.logo === logoClassic) {
    this.setState({ logo: logoSpace })
  }
  else {
    this.setState({ logo: logoClassic })
  }
}

render() {
  return (
    <div style={ sideStyle }>
      <div style={ logoStyle } >
        <Link to="/">
          <img src={this.state.logo} style={ imageStyle } alt="logo" />
        </Link>
      </div>
      <div>
        <Button style={ buttonStyle } variant="info"
          onClick={ this.props.onThemeClick && this.changeTheme }>Ändra tema</Button>{' '}
        <Button style={ buttonStyle } variant="info" onClick={ reloadPage }>Slumpa</Button>{' '}
        <Link to="/marsvin">
          <Button style={ buttonStyle } variant="info">Marsvin</Button>{' '}
        </Link>
      </div>
    </div>
  )
}
```

# REFLEKTIONER

- Arbetssätt - Wireframing, GitHub issues & grundkodning utfördes gemensamt
- KAOS - ändå lärorikt? Idén fick omarbetas mitt i projektet
- "Huvuddelen" av strukturen av issues kunde återanvändas
- React svårt att greppa, men kraftfullt ramverk
- Typning både hjälp men också en extra "dimension" att tänka på
- Gott samarbete, både delat upp men suttit mycket tillsammans och hjälpts åt

The background is a blue gradient with abstract white lines resembling circuit traces or data paths in the corners. These lines connect small circles, some of which are larger than others, creating a network-like structure. The lines are more prominent in the top-left and bottom-left corners, and less so in the top-right and bottom-right corners.

LIVE DEMO