

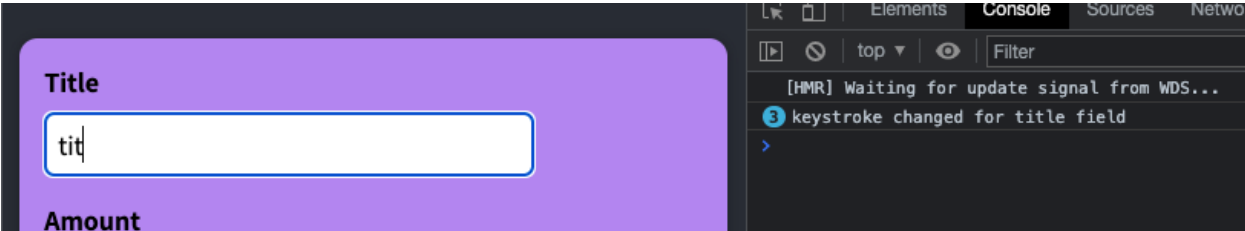
18. Listening to user input in a form

- Lets say, that we have to listen to each and every keystroke that is entered in the input field
 - This is helpful for searching where the search results gets updated even for every single change in the input field instead of showing results only after submitting the search form
- How can we do this
- We can use eventlisteners for every input fields
 - We can use onChange or onInput prop, both are same but onChange has some advantages over onInput, that is onChange works fine for all type of inputs

```
4  const ExpenseForm={()=>{
5    return (
6      <form>
7        <div className='new-expense__controls'>
8          <div className="new-expense__control">
9            <label>Title</label>
10           <input type='text' onChange={} />
11         </div>
12
13         <div className="new-expense__control">
14           <label>Amount</label>
```

-
- Now we have to give in a function for onChange prop
 - In the below screenshot, we have defined a function and specified it inside the onChange prop

```
4  const ExpenseForm={()=>{
5    const titleChangeHandler={()=>{
6      console.log("keystroke changed for title field");
7    }
8    return (
9      <form>
10        <div className='new-expense__controls'>
11          <div className="new-expense__control">
12            <label>Title</label>
13            <input type='text' onChange={titleChangeHandler} />
14          </div>
15
```

-
- 

- From this we could see that for every change in the field, the titlechangehandler function is evaluated

- Now the question is, how could we get the value that the user enters immediately after each keystroke instead of submitting the search form
 - In normal vanilla js:

```
document.getElementById('').addEventListener('click', (event) => {})
```

- Here in vanilla js this is the way we use eventlisteners, where every event has some values and properties, in this, click event has some properties which is then passed to the function
- Similarly we can use it in our react eventlistener function also

```
<input type='text' onChange={titleChangeHandler} />
```

- Here the onChange event passes some event properties to the titlechangehandler function, so let us accept a argument to access the event object in the titlechangehandlerfunction

```
4  const ExpenseForm=()=>{
5    const titleChangeHandler=(event)=>{
6      console.log(event);
7    }
8    return (
9      <form>
10        <div className='new-expense__controls'>
11          <div className='new-expense__control'>
12            <label>Title</label>
13            <input type='text' onChange={titleChangeHandler} />
14          </div>
15        </div>
16      </form>
17    )
18  }
```

- Here let us get the event as an argument and let us print that in our console

Title

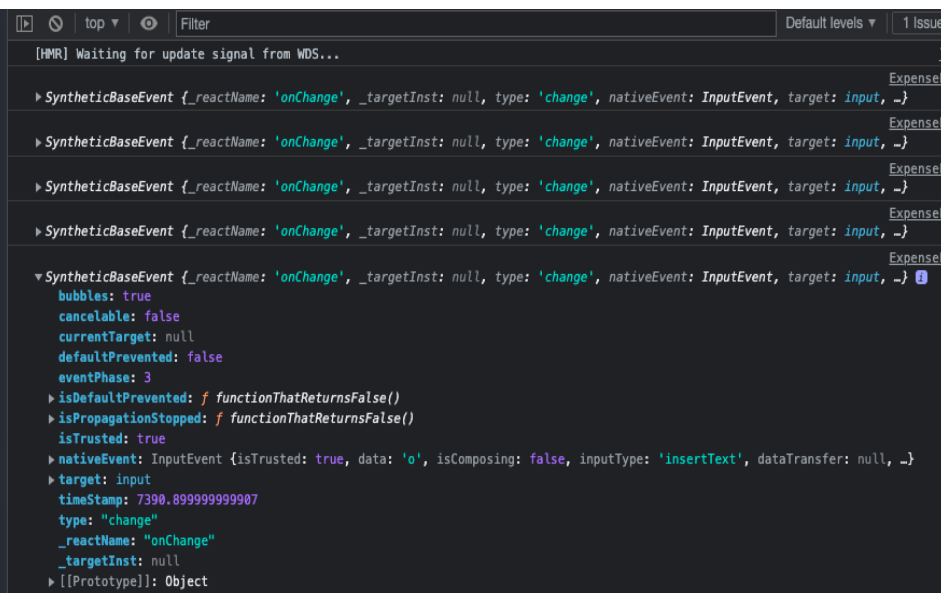
Amount

Date

Add Expense

April 2022

\$432



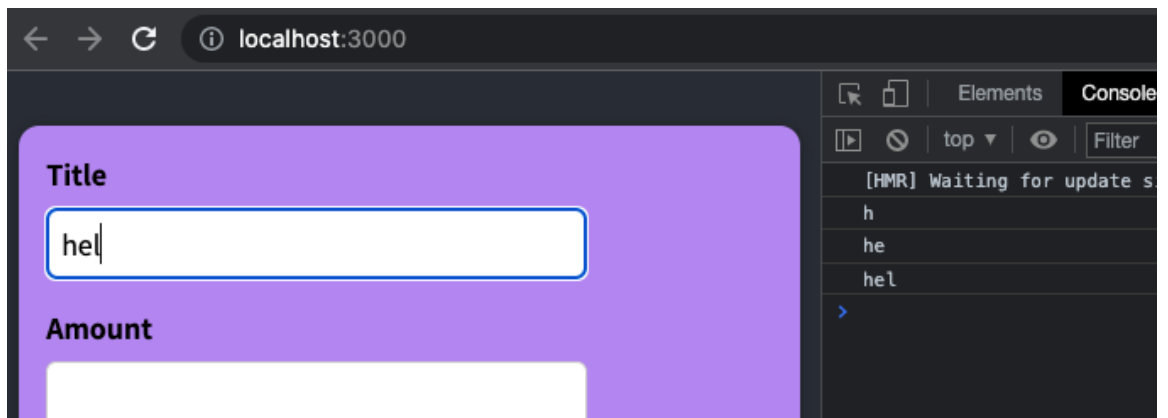
- Here this event object is displayed in our log which has several properties, here let us expand the target dropdown

```
▼ SyntheticBaseEvent {_reactName: 'onChange',
  bubbles: true
  cancelable: false
  currentTarget: null
  defaultPrevented: false
  eventPhase: 3
  ▶ isDefaultPrevented: f functionThatReturns...
  ▶ isPropagationStopped: f functionThatReturn...
  isTrusted: true
  ▶ nativeEvent: InputEvent {isTrusted: true,
    target: input
    ▼ target: input
      value: "hello"
      ▶ __reactEvents$1lz6o29o3il: Set(1) {'inve...
      ▶ __reactFiber$1lz6o29o3il: FiberNode {tag...
      ▶ __reactProps$1lz6o29o3il: {type: 'text',...
      ▶ _valueTracker: {getValue: f, setValue: j...
      ▶ _wrapperState: {initialChecked: undefin...
```

- Here we could get the value that is entered in the input field immediately after each keystroke
- We don't need the entire event property, we just need the value
- So

```
4  const ExpenseForm={()=>{
5    const titleChangeHandler=(event)=>{
6      console.log(event.target.value);
7    }
8    return (
```

○



○

- Now we were able to listen each and every keystroke in our form field