Week 2

Week2

- Week2
  - Main Goal
  - API on Documentation
  - Code based API methods
    - Result

Main Goal

- Find API methods related to ReactJs
- Design queries to collect the dataset
- Analyse the results and visualize them

API on Documentation

```
React.Component
->
constructor()
static getDerivedStateFromProps()
render()
componentDidMount()

getDerivedStateFromProps()
shouldComponentUpdate()
render()
getSnapshotBeforeUpdate()
componentDidUpdate()

componentWillUnmount()

componentDidCatch()

setState()
forceUpdate()

defaultProps
displayName

props
state



React.PureComponent
createElement()
createFactory()
cloneElement()
isValidElement()
React.Children
React.Fragment
React.createRef
React.forwardRef
```

```
ReactDOM
->
render()
hydrate()
unmountComponentAtNode()
findDOMNode()
createPortal()

ReactDOMServer
->
renderToString()
renderToStaticMarkup()
renderToNodeStream()
renderToStaticNodeStream()


DOM Elements
->
No API functions,but Attributes

SyntheticEvent
->
No API functions,but Attributes

Test Utilities
->
Simulate
renderIntoDocument()
mockComponent()
isElement()
isElementOfType()
isDOMComponent()
isCompositeComponent()
isCompositeComponentWithType()
findAllInRenderedTree()
scryRenderedDOMComponentsWithClass()
findRenderedDOMComponentWithClass()
scryRenderedDOMComponentsWithTag()
findRenderedDOMComponentWithTag()
scryRenderedComponentsWithType()
findRenderedComponentWithType()


Shallow Renderer
->
shallowRenderer.render()
shallowRenderer.getRenderOutput()

Test Renderer
->
TestRenderer.create()
testRenderer.toJSON()
testRenderer.toTree()
testRenderer.update()
testRenderer.unmount()
testRenderer.getInstance()
testRenderer.root
testInstance.find()
testInstance.findByType()
testInstance.findByProps()
testInstance.findAll()
testInstance.findAllByType()
testInstance.findAllByProps()
testInstance.instance
testInstance.type
testInstance.props
testInstance.parent
testInstance.children
```

Code based API methods

```
  <!-- ... other HTML ... -->

  <!-- Load React. -->
  <!-- Note: when deploying, replace "development.js" with "production.min.js". -->
  <script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>
  <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin>
</script>

  <!-- Load our React component. -->
  <script src="like_button.js"></script>

</body>
```

Result

```
ChildReconciler
Class
Component
ComponentDummy
FiberNode
PropertyInfoRecord
PureComponent
ReactBatch
ReactRoot
ReactWork
SyntheticEvent
accumulateDirectDispatches
accumulateDirectDispatchesSingle
accumulateDirectionalDispatches
accumulateDispatches
accumulateEnterLeaveDispatches
accumulateInto
accumulateTwoPhaseDispatches
accumulateTwoPhaseDispatchesSingle
accumulateTwoPhaseDispatchesSingleSkipTarget
accumulateTwoPhaseDispatchesSkipTarget
addEventBubbleListener
addEventCaptureListener
addEventPoolingTo
addEventTypeNameToConfig
addRootToSchedule
adoptClassInstance
appendAllChildren
appendChild
appendChildToContainer
appendInitialChild
appendUpdateToQueue
applyDerivedStateFromProps
assertIsMounted
assertValidProps
assignFiberPropertiesInDEV
bailoutOnAlreadyFinishedWork
bailoutOnLowPriority
batchedUpdates
batchedUpdates$1
beginWork
cacheContext
callCallback
callComponentWillMount
callComponentWillReceiveProps
```

```
callback
camelize
camelizeStyleName
canHydrateInstance
canHydrateTextInstance
cancelWorkTimer
captureCommitPhaseError
catchErrors
ceiling
checkActualRenderTimeStackEmpty
checkClassInstance
checkHasForceUpdateAfterProcessing
checkPropTypes
checkSelectPropTypes
checkShouldComponentUpdate
checkThatStackIsEmpty
cloneAndReplaceKey
cloneChildFibers
cloneElement
cloneElementWithValidation
cloneUpdateQueue
coerceRef
commitAllHostEffects
commitAllLifeCycles
commitAttachRef
commitBeforeMutationLifeCycles
commitBeforeMutationLifecycles
commitContainer
commitDeletion
commitDetachRef
commitLifeCycles
commitMount
commitNestedUnmounts
commitPlacement
commitResetTextContent
commitRoot
commitTextUpdate
commitUnmount
commitUpdate
commitUpdateQueue
commitWork
completeRoot
completeUnitOfWork
completeWork
computeAsyncExpiration
computeExpirationBucket
computeExpirationForFiber
computeInteractiveExpiration
computeUniqueAsyncExpiration
constructClassInstance
constructSelectEvent
containsNode
countChildren
createAndAccumulateChangeEvent
createCapturedValue
createChild
createClassErrorUpdate
createContainer
createContext
createCursor
createDangerousStringForStyles
createElement
createElement$1
createElementWithValidation
createFactoryWithValidation
createFiberFromElement
createFiberFromFragment
createFiberFromHostInstanceForDeletion
```

```
createFiberFromPortal
createFiberFromProfiler
createFiberFromText
createFiberRoot
createHostRootFiber
createInstance
createPortal
createPortal$1
createRef
createRoot
createRootErrorUpdate
createTextInstance
createTextNode$1
createUpdate
createUpdateQueue
createWorkInProgress
dangerousStyleValue
deferredUpdates
defineKeyPropWarningGetter
defineRefPropWarningGetter
deleteChild
deleteHydratableInstance
deleteRemainingChildren
describeFiber
detachFiber
detachTracker
didNotFindHydratableContainerInstance
didNotFindHydratableContainerTextInstance
didNotFindHydratableInstance
didNotFindHydratableTextInstance
didNotHydrateContainerInstance
didNotHydrateInstance
didNotMatchHydratedContainerTextInstance
didNotMatchHydratedTextInstance
diffHydratedProperties$1
diffHydratedText$1
diffProperties$1
dispatch
dispatchEvent
dispatchInteractiveEvent
emptyFunction
emptyPortalContainer
enqueueCapturedUpdate
enqueueStateRestore
enqueueUpdate
ensureListeningTo
ensureWorkInProgressQueueIsAClone
enterHydrationState
escape
escapeUserProvidedKey
executeDispatch
executeDispatchesInOrder
expirationTimeToMs
extractBeforeInputEvent
extractCompositionEvent
extractEvents
finalizeInitialChildren
findCurrentFiberUsingSlowPath
findCurrentHostFiber
findCurrentHostFiberWithNoPortals
findCurrentUnmaskedContext
findHighestPriorityRoot
findHostInstance
findHostInstanceWithNoPortals
findNextPendingPriorityLevel
findRootContainerNode
finishClassComponent
finishRendering
```

```
flattenChildren
flushControlled
flushInteractiveUpdates$1
flushRoot
flushSync
forEachAccumulated
forEachChildren
forEachSingleChild
forwardRef
get
getActiveElement
getChildHostContext
getChildNamespace
getClosestInstanceFromNode
getCommitTime
getComponentKey
getComponentName
getCompositionEventType
getContextChangedBits
getContextCurrentValue
getContextForSubtree
getCurrentComponentErrorInfo
getCurrentFiberOwnerName$1
getCurrentFiberStackAddendum$1
getData
getDataFromCustomEvent
getDeclarationErrorAddendum
getEventCharCode
getEventKey
getEventModifierState
getEventTarget
getFallbackBeforeInputChars
getFiberCurrentPropsFromNode$1
getFiberTagFromObjectType
getFirstHydratableChild
getHostContext
getHostParentFiber
getHostProps
getHostProps$1
getHostProps$2
getHostProps$3
getHostSibling
getInstIfValueChanged
getInstanceFromNode$1
getIntrinsicNamespace
getIteratorFn
getLeafNode
getListener
getListeningForDocument
getLowestCommonAncestor
getMaskedContext
getModernOffsetsFromPoints
getNativeBeforeInputChars
getNextHydratableSibling
getNodeForCharacterOffset
getNodeFromInstance$1
getOffsets
getOwnerDocumentFromRootContainer
getParent
getParentInstance
getPooledEvent
getPooledTraverseContext
getPooledWarningPropertyDefinition
getPossibleStandardName
getPropertyInfo
getPublicInstance
getPublicRootInstance
getRawEventName
```

```
getReactRootElementInContainer
getRootHostContainer
getRootHostContext
getSafeValue
getSelection
getSelection$1
getSelectionInformation
getSiblingNode
getSourceInfoErrorAddendum
getStackAddendum
getStackAddendum$1
getStackAddendum$2
getStackAddendumByWorkInProgressFiber
getStateFromUpdate
getTargetInstForChangeEvent
getTargetInstForClickEvent
getTargetInstForInputEventPolyfill
getTargetInstForInputOrChangeEvent
getText
getTextContentAccessor
getTopLevelCallbackBookKeeping
getTracker
getUnmaskedContext
getValueForAttribute
getValueForProperty
getValueFromNode
getVendorPrefixedEventName
handleControlledInputBlur
handleEventsForInputEventPolyfill
handlePropertyChange
handleTopLevel
has
hasContextChanged
hasSelectionCapabilities
hasValidKey
hasValidRef
hydrateInstance
hydrateTextInstance
hyphenate
hyphenateStyleName
initWrapperState
initWrapperState$1
initWrapperState$2
initialize
injectEventPluginOrder
injectEventPluginsByName
injectInternals
injectIntoDevTools
insertBefore
insertInContainerBefore
insertNonHydratedInstance
interactiveUpdates
interactiveUpdates$1
invalidateContextProvider
invariant
is
isAlreadyFailedLegacyErrorBoundary
isAttributeNameSafe
isCheckable
isContextConsumer
isContextProvider
isControlled
isCustomComponent
isEnabled
isEventSupported
isFallbackCompositionEnd
isFallbackCompositionStart
isFiberMounted
```

```
isFiberMountedImpl
isHostParent
isInDocument
isInteractive
isKeypressCommand
isListeningToAllDependencies
isMounted
isNode
isTextInputElement
isTextNode
isValidContainer
isValidElement
isValidElementType
legacyCreateRootFromDOMContainer
legacyRenderSubtreeIntoContainer
listenTo
listenerAtPhase
logCapturedError
logError
makeEmptyFunction
makePrefixMap
manualDispatchChangeEvent
mapChildren
mapIntoWithKeyPrefixInternal
mapRemainingChildren
mapSingleChildIntoContext
markActualRenderTimeStarted
markCommittedPriorityLevels
markLegacyErrorBoundaryAsFailed
markPendingPriorityLevel
markPingedPriorityLevel
markRef
markRef$1
markSuspendedPriorityLevel
markUpdate
memoizeProps
memoizeState
modifierStateGetter
mountClassInstance
mountIndeterminateComponent
msToExpirationTime
needsStateRestore
onBlock
onCommitRoot
onCommitUnmount
onError
onUncaughtError
onlyChild
pauseActualRenderTimerIfRunning
performAsyncWork
performSyncWork
performUnitOfWork
performWork
performWorkOnRoot
placeChild
placeSingleChild
pop
popContextProvider
popHostContainer
popHostContext
popHydrationState
popProvider
popToNextHostParent
popTopLevelContextObject
postMountWrapper
postMountWrapper$1
postMountWrapper$2
postMountWrapper$3
```

```
postUpdateWrapper
precacheFiberNode
prefixKey
prepareForCommit
prepareToHydrateHostInstance
prepareToHydrateHostTextInstance
prepareUpdate
printWarning
processChildContext
processUpdateQueue
propagateContextChange
publishEventForPlugin
publishRegistrationName
push
pushContextProvider
pushHostContainer
pushHostContext
pushHostRootContext
pushProvider
pushTopLevelContextObject
recalculateCurrentTime
recomputePluginOrdering
reconcileChildFibers
reconcileChildren
reconcileChildrenArray
reconcileChildrenAtExpirationTime
reconcileChildrenIterator
reconcileSingleElement
reconcileSinglePortal
reconcileSingleTextNode
recordCommitTime
recordEffect
recordElapsedActualRenderTime
recordElapsedBaseRenderTimeIfRunning
recordScheduleUpdate
releasePooledEvent
releaseTopLevelCallbackBookKeeping
releaseTraverseContext
removeChild
removeChildFromContainer
renderRoot
requestRetry
requestWork
requiredContext
reset
resetActualRenderTimer
resetAfterCommit
resetCurrentFiber
resetExpirationTime
resetHasForceUpdateBeforeProcessing
resetHydrationState
resetStack
resetStackAfterFatalErrorInDev
resetTextContent
restoreControlledState
restoreControlledState$1
restoreControlledState$2
restoreControlledState$3
restoreSelection
restoreStateIfNeeded
restoreStateOfTarget
resumeActualRenderTimerIfPaused
resumeMountClassInstance
retrySuspendedRoot
reuseChildrenEffects
runEventInBatch
runEventsInBatch
runExtractedEventsInBatch
```

```
safelyCallComponentWillUnmount
safelyDetachRef
scheduleCallbackWithExpiration
schedulePing
scheduleRootUpdate
scheduleWork$1
set
setCurrentFiber
setCurrentPhase
setDefaultValue
setEnabled
setInitialDOMProperties
setInitialProperties$1
setOffsets
setSelection
setValueForProperty
setValueForStyles
shallowEqual
shim
shouldAutoFocusHostComponent
shouldConstruct
shouldDeprioritizeSubtree
shouldHydrateDueToLegacyHeuristic
shouldIgnoreAttribute
shouldPreventMouseEvent
shouldRemoveAttribute
shouldRemoveAttributeWithWarning
shouldSetTextContent
shouldUseChangeEvent
shouldUseClickEvent
shouldUseNative
shouldYield
showErrorDialog
startBaseRenderTimer
startCommitHostEffectsTimer
startCommitLifeCyclesTimer
startCommitSnapshotEffectsTimer
startCommitTimer
startPhaseTimer
startRequestCallbackTimer
startWatchingForValueChange
startWorkLoopTimer
startWorkTimer
stopBaseRenderTimerIfRunning
stopCommitHostEffectsTimer
stopCommitLifeCyclesTimer
stopCommitSnapshotEffectsTimer
stopCommitTimer
stopFailedWorkTimer
stopPhaseTimer
stopRequestCallbackTimer
stopWatchingForValueChange
stopWorkLoopTimer
stopWorkTimer
suspendRoot
syncUpdates
throwException
throwOnInvalidObjectType
toArray
toObject
track
trackValueOnNode
trapBubbledEvent
trapCapturedEvent
trapClickOnNonInteractiveElement
traverseAllChildren
traverseAllChildrenImpl
traverseEnterLeave
```

```
  traverseTwoPhase
  tryHydrate
  tryToClaimNextHydratableInstance
  unbatchedUpdates
  unmountHostComponents
  unsafeCastDOMTopLevelTypeToString
  unsafeCastStringToDOMTopLevelType
  unwindInterruptedWork
  unwindWork
  updateChecked
  updateClassComponent
  updateClassInstance
  updateContainer
  updateContainerAtExpirationTime
  updateContextConsumer
  updateContextProvider
  updateDOMProperties
  updateElement
  updateFiberProps
  updateForwardRef
  updateFragment
  updateFromMap
  updateFunctionalComponent
  updateHostComponent
  updateHostRoot
  updateHostText
  updateMode
  updateNamedCousins
  updateOptions
  updatePortal
  updatePortalComponent
  updateProfiler
  updateProperties$1
  updateSlot
  updateTextNode
  updateTimeoutComponent
  updateValueIfChanged
  updateWrapper
  updateWrapper$1
  useFiber
  validateChildKeys
  validateExplicitKey
  validateFormat
  validateFragmentProps
  validatePropTypes
  validateProperties
  validateProperties$1
  validateProperties$2
  validateProperty
  validateProps
  warn
  warnForDeletedHydratableElement$1
  warnForDeletedHydratableText$1
  warnForInsertedHydratedElement$1
  warnForInsertedHydratedText$1
  warnForUnmatchedText$1
  warnInvalidARIAProps
  warnNoop
  warnOnFunctionType
  warnOnInvalidKey
  warning
  workLoop
```

Code:

```
import java.io.*;
import java.util.Scanner;
import java.util.TreeSet;

public class ReactAPI {
    public static TreeSet<String> functions = new TreeSet<>();
    public static boolean isWord(char c){
        if(c>='a' && c<='z' || c>='A' && c<='Z'){
            return true;
        }
        return false;
    }

    public static void handleFile(String filename) throws FileNotFoundException {
        File js1 = new File(filename);
        BufferedReader reader = null;
        reader = new BufferedReader(new FileReader(js1));
        Scanner in = new Scanner(reader);
        while (in.hasNextLine()){
            String str = in.nextLine();
//            System.out.println("Line:"+str);
            if(str.contains("function")){
                int posi = str.indexOf("function");
//                int last = str.lastIndexOf("function");
//                if(posi!=last)System.out.println("ERROR");//YES!!! It appears.
                Scanner sin = new Scanner(str);
                while (sin.hasNext()){
                    String token = sin.next();
                    if(token.equals("function")){
                        if(!sin.hasNext())break;
                        token=sin.next();
                        if(token.contains("(")){
                            String name = token.substring(0,token.indexOf('('));
                            if(name.length()!=0){
                                functions.add(name);
                                System.out.println("ADD:"+name);
                            }
                        }
                    }
                }
//                int nxtToken = posi+9;
//                String name = "";
//                int sl = str.length();
//
//                while (isWord(str.charAt(nxtToken))){
//                    name = name+str.charAt(nxtToken);
//                    nxtToken++;
//                    if(nxtToken>=sl){
//                        name="";
//                        break;
//                    }
//                }
//                if(!name.equals("") && str.charAt(nxtToken)=='('){
//                    functions.add(name);
//                    System.out.println("ADD:"+name);
//                }


            }
        }


    }

    public static void main(String args[]) throws IOException {
        functions.clear();
        try {
//            handleFile("test.txt");
```

```
            handleFile("react.development.js");
            handleFile("react-dom.development.js");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        File output = new File("ReactAPIFunctions.txt");
        FileWriter fw = null;
        fw = new FileWriter(output);
        for(String str:functions){
            System.out.println(str);
            fw.write(str+"\r\n");
        }
        fw.flush();
        fw.close();

    }

}
```