

LOAD DATA

```

Route '/', component: BranchesAppComponent

BranchesAppComponent.ts
  async created() {
    this.init()
  }
  ...
  methods: {
    init() {
      const app = myContainer.get<IBranchesApp>(TYPES.IBranchesApp) // will inject
      the correct databases, localStorage, localStorageRetrievers etc.
      app.setURL(getWindowURI())
      app.init()
    }
  }

BranchesApp.ts
  init() {
    this.datastore.init() // DataStore has LocalStorageHandler already injected into it
    this.ui_bridges.subscribe(this.datastore)
    this.getContentItem
  }

UIBridges.ts
  TreeUIBridge
  ContentItemUIBridge

  subscribe(subscribable: ISubscribable)
  {
    subscribable.onUpdate(this.handleDataUpdate)
  }
  handleDataUpdate((dataUpdate: IDataUpdate)) {
    switch (dataUpdate.datatype) {
      case DATA_TYPES.TREES:
        treeUIBridge.handleDataUpdate((dataUpdate))
      case DATA_TYPES.TREE_USER_DATA:
        treeUserDataUIBridge.handleDataUpdate((dataUpdate))
    }
  }

TreeUIBridge.ts {
  handleDataUpdate((dataUpdate: IDataUpdate)) {
    const sigmaNodeId = DATA_SIGMA_MAP.getFromTreeId(dataUpdate.treeld)
    sigmaHandler.update({sigmaNodeId: dataUpdate.treeld, updateType:
DATA_TYPES.TREES, newTreeData: dataUpdate.val})
  }
}
TreeUserDataUIBridge.ts {
  handleDataUpdate((dataUpdate: IDataUpdate)) {
    const sigmaNodeId = DATA_SIGMA_MAP.getFromTreeId(dataUpdate.treeld)
    sigmaHandler.update({sigmaNodeId: dataUpdate.treeld, updateType:
DATA_TYPES.TREE_USER_DATA, newTreeUserData: dataUpdate.val})
  }
}

ContentItemUserDataUIBridge.ts {
  handleDataUpdate((dataUpdate: IDataUpdate)) {
    const sigmaNodeIds = DATA_SIGMA_MAP.getFromContentId(dataUpdate.contentId)
    sigmaNodeIds.forEach(sigmaNodeId => {
      sigmaHandler.update({sigmaNodeId: dataUpdate.treeld, updateType:
DATA_TYPES.TREE_USER_DATA, newTreeUserData: dataUpdate.val})
    })
  }
}

DataStore.ts() {
  graph = (nodes: [], edges: []) /
```

SubscribeToFirebase class
-subscribes to updates from firebase ref
-transforms those updates and publishes them to any subscribers
-has an onUpdate method to register subscribers
>

ACTIONS

Examples

new fact as child
under a category

userInteraction

edit fact on tree

dragNode

```
CreateTree
createContentItem
CreateExercise
```



