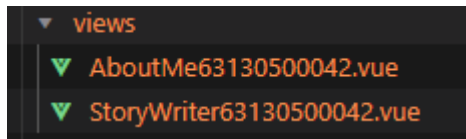


1.

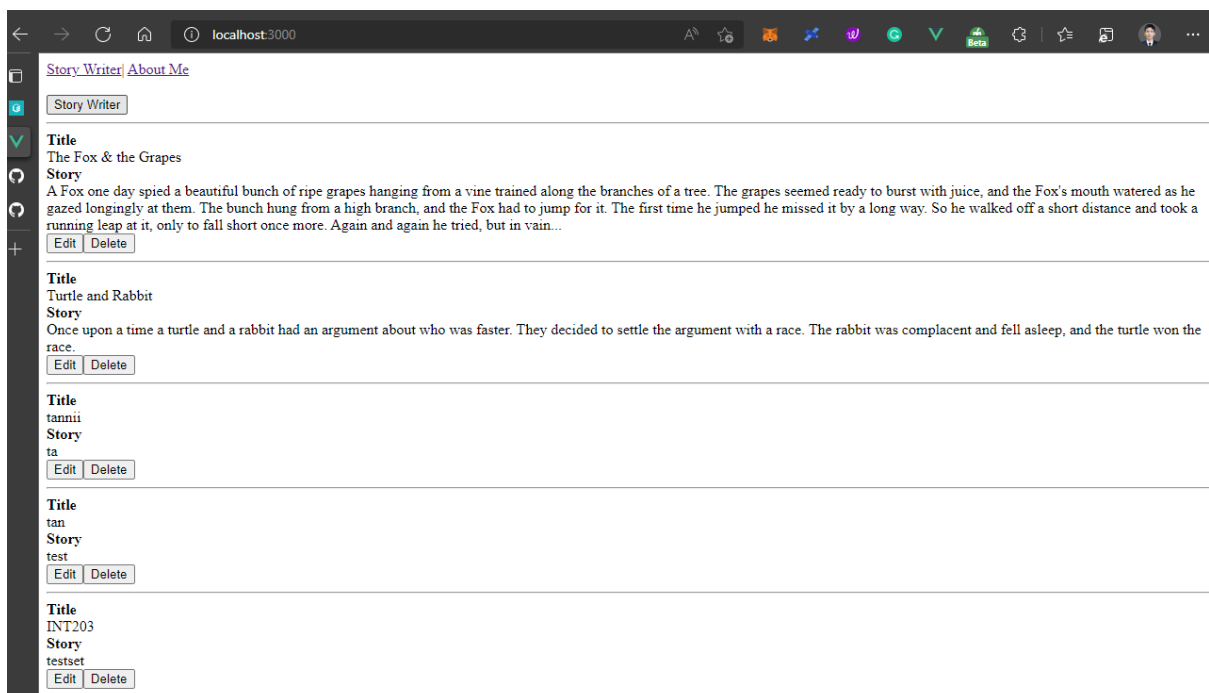


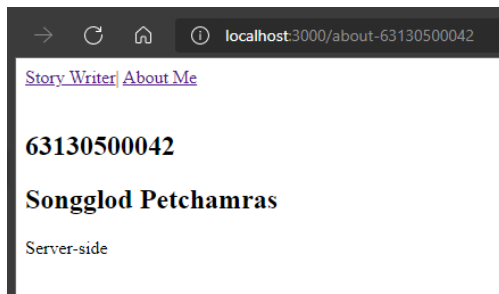
```
import { createRouter, createWebHistory } from "vue-router";
import AboutMe from "../views/AboutMe63130500042.vue";
import StoryWriter from "../views/StoryWriter63130500042.vue";

const history = createWebHistory();
const routes = [
  {
    path: "/", // homepage
    name: "StoryWriter63130500042",
    component: StoryWriter,
  },
  {
    path: "/about-63130500042",
    name: "About63130500042",
    component: AboutMe,
  },
];

const router = createRouter({ history, routes });

export default router;
```





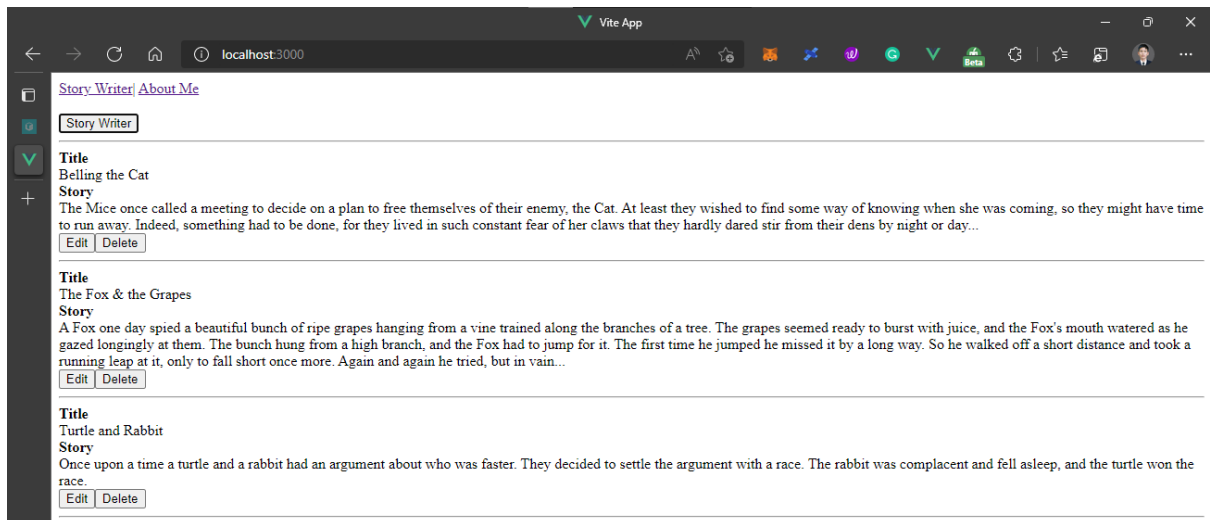
## 2.1

```
// GET
const getStoryList = async (url) => {
  const res = await fetch(url);
  if (res.status === 200) {
    stories.value = await res.json();
  }
};
```

```
<script setup>
const props = defineProps({
  stories: {
    type: [Object, Array],
  },
});
const emits = defineEmits(["edit", "delete"]);
</script>

<template>
  <div id="story-list">
    <div v-for="detail in stories" :key="detail.id">
      <div style="font-weight: bold">Title</div>
      <div>{{ detail.title }}</div>
      <div style="font-weight: bold">Story</div>
      <div>{{ detail.story }}</div>
      <button @click.left="$emit('edit', detail)">Edit</button>
      <button @click.left="$emit('delete', detail.id)">Delete</button>
      <hr />
    </div>
  </div>
</template>

<style scoped></style>
```



2.2



```

<script setup>
import { ref, computed } from "vue";
const props = defineProps({
  editStory: { type: Object },
});
const emits = defineEmits(["add", "update"]);

const isClick = ref(false);

const whenClick = () => {
  isClick.value = isClick.value ? false : true;
};

const newStory = computed(() => {
  return {
    id: props.editStory.id,
    title: props.editStory.title,
    story: props.editStory.story,
  };
});
</script>

```

```

<button type="button" @click.left="whenClick" :disabled="isClick">
  Story Writer
</button>

```

← → ↻ 🏠 ⓘ localhost:3000 🔊 ⚙️ 🦊

[Story Writer](#) [About Me](#)

Story Writer

---

## Story Writer

Title

Story

Create Cancel

→ ↻ 🏠 ⓘ localhost:3000 🔊 ⚙️ 🦊 🌐

[Story Writer](#) [About Me](#)

Story Writer

---

## Story Writer

Title

Story

Create Cancel

---

**Title**  
test  
**Story**  
test

Edit Delete

Vite App

localhost:3000

[Story Writer](#) [About Me](#)

Story Writer

# Story Writer

Title

test

Story

test

CreateCancel

Title

test

Story

test

EditDelete

localhost:3000

[Story Writer](#) [About Me](#)

Story Writer

Title

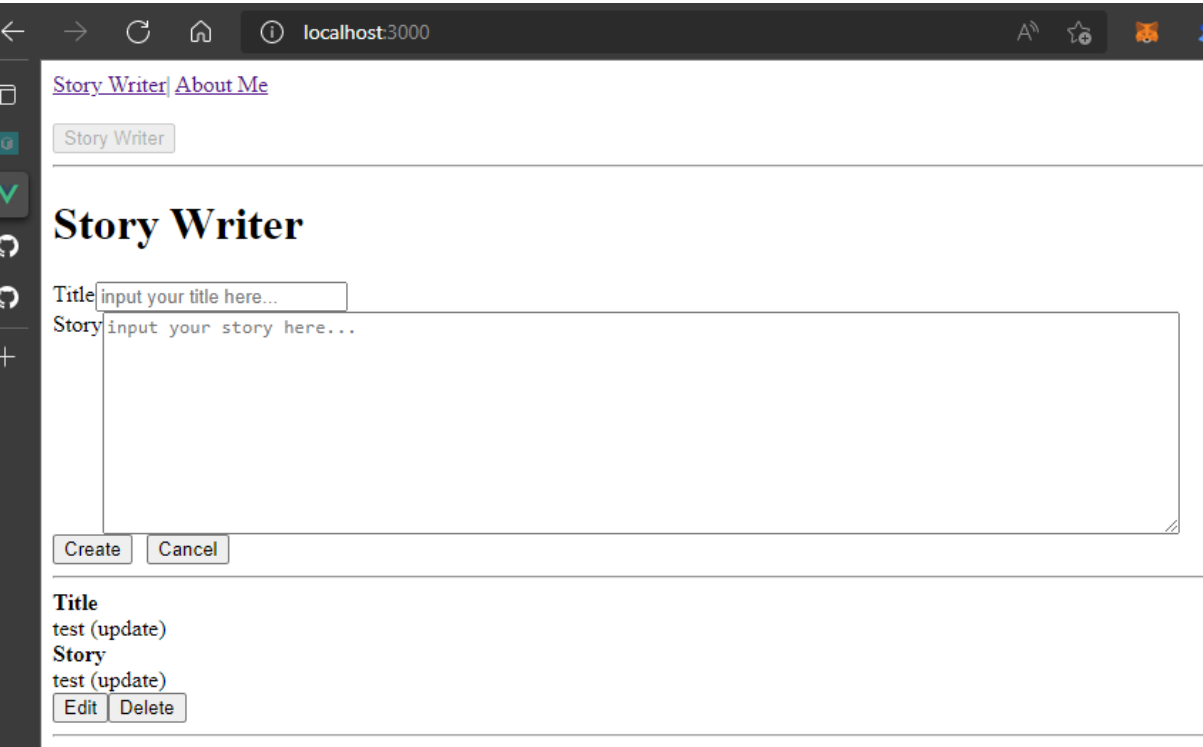
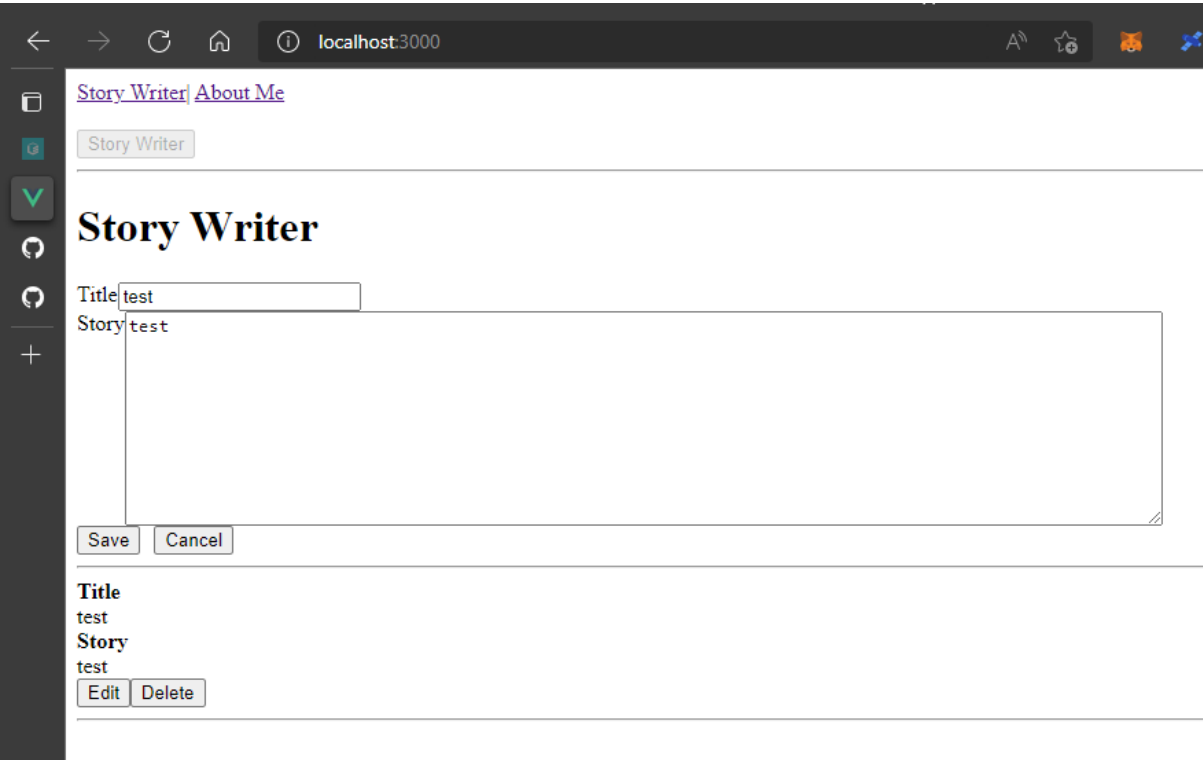
test

Story

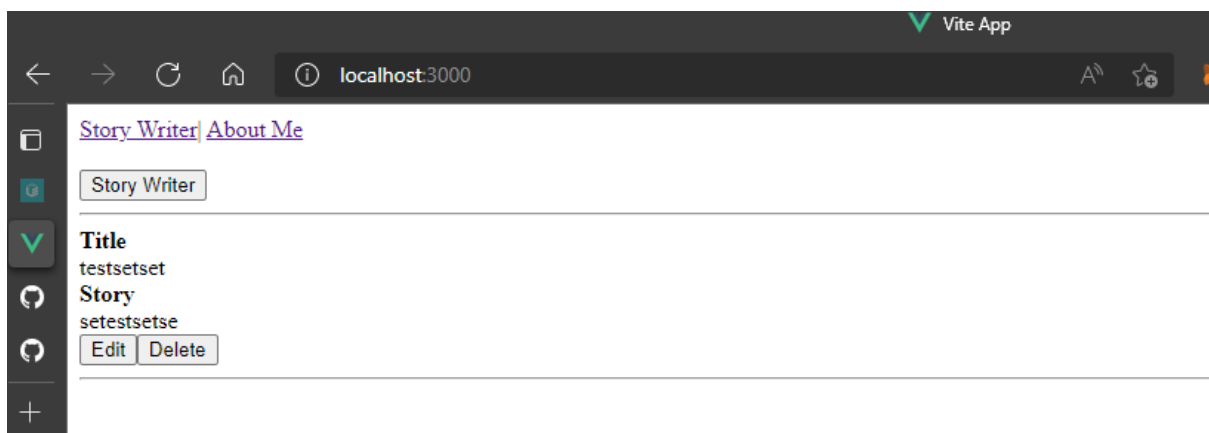
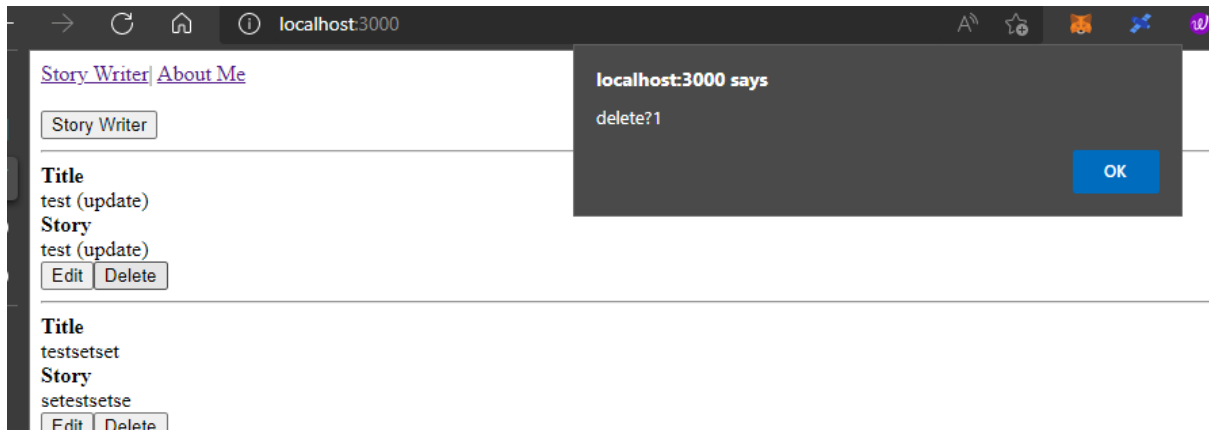
test

EditDelete

2.3



## 2.4



script

```
<script setup>
import AddStory from "../components/AddStory63130500042.vue";
import Story from "../components/Story63130500042.vue";
import { onBeforeMount, ref } from "vue";

const URL = "http://localhost:5000/stories";

const stories = ref([]); // for GET
const editStory = ref({});
```

```
// GET
const getStoryList = async (url) => {
  const res = await fetch(url);
  if (res.status === 200) {
    stories.value = await res.json();
  }
};
```

```
// POST
const createStory = async (new_story) => {
  const res = await fetch(URL, {
    method: "POST",
    headers: {
      "content-type": "application/json",
    },
    body: JSON.stringify(new_story),
  });
  if (res.status === 201) {
    const temp = await res.json();
    stories.value.push(temp);
  }
};
```

```
// DELETE
const deleteStory = async (id) => {
  const res = await fetch(URL + "/" + id, { method: "DELETE" });
  if (res.status === 200) {
    alert("delete?" + id);
    stories.value = stories.value.filter((story) => story.id !== id);
    console.log("deleted successfully");
  } else {
    console.log("deleted failed");
  }
};

const toEdit = (story) => (editStory.value = story);
```



```
// PUT
const updateStory = async (story) => {
  const res = await fetch(URL + "/" + story.id, {
    method: "PUT",
    headers: {
      "content-type": "application/json",
    },
    body: JSON.stringify({ title: story.title, story: story.story }),
  });
  if (res.status === 200) {
    const editedStory = await res.json();
    stories.value = stories.value.map((story) =>
      story.id === editedStory.id
        ? { title: editedStory.title, story: editedStory.story }
        : story
    );
  }
  editStory.value = "";
};
```

```
<template>
  <div id="story-writer">
    <AddStory
      @add="createStory"
      @update="updateStory"
      :editStory="editStory"
    />
    <Story :stories="stories" @edit="toEdit" @delete="deleteStory" />
  </div>
</template>
```

```

<script setup>
import { ref, computed } from "vue";
const props = defineProps({
  editStory: { type: Object },
});
const emits = defineEmits(["add", "update"]);

const isClick = ref(false);

const whenClick = () => {
  isClick.value = isClick.value ? false : true;
};

const newStory = computed(() => {
  return {
    id: props.editStory.id,
    title: props.editStory.title,
    story: props.editStory.story,
  };
});
</script>

```

```

<div>
  <button
    v-if="newStory.id > 0"
    style="margin-right: 10px"
    @click.Left="$emit('update', newStory)"
  >
    Save
  </button>
  <button
    v-else
    style="margin-right: 10px"
    @click.Left="$emit('add', newStory)"
  >
    Create
  </button>
  <button @click.Left="whenClick">Cancel</button>
</div>

```

```
<script setup>
const props = defineProps({
  stories: {
    type: [Object, Array],
  },
});
const emits = defineEmits(["edit", "delete"]);
</script>

<template>
  <div id="story-list">
    <div v-for="detail in stories" :key="detail.id">
      <div style="font-weight: bold">Title</div>
      <div>{{ detail.title }}</div>
      <div style="font-weight: bold">Story</div>
      <div>{{ detail.story }}</div>
      <button @click.left="$emit('edit', detail)">Edit</button>
      <button @click.left="$emit('delete', detail.id)">Delete</button>
      <hr />
    </div>
  </div>
</template>

<style scoped></style>
```