File: ./package-lock.json

```
[binary]
```

File: ./src/views/SignIn.vue

```
<template>
  <v-container fill-height fluid class="primary">
    <v-card class="mx-auto pa-7 rounded-xl" width="500">
      <v-card-title>
        <v-row class="text-h4 mb-2 text-break text-center" justify="center">
          DOUTOR INTELIGENTE
        </v-row>
        <v-row justify="center">
          <v-img
            src="../assets/logo.png"
            contain
            max-width="150"
            max-height="150"
          />
        </v-row>
      </v-card-title>

      <v-card-text class="my-auto mt-5">
        <v-form
          ref="form"
          lazy-validation
          v-model="valid"
          @keyup.native.enter="login"
        >
          <v-text-field
            v-model="key"
            label="Chave de acesso"
            :rules="[non_empty_rule]"
            outlined
            clearable
            clear-icon="fas fa-times"
            color="secondary"
            prepend-icon="fas fa-key"
          ></v-text-field>

          <v-file-input
            v-model="file"
            :rules="[non_empty_rule]"
            accept="image/*"
            prepend-icon="fas fa-images"
            label="Radiografia"
            outlined
            clear-icon="fas fa-times"
            color="secondary"
          ></v-file-input>
        </v-form>

        <v-row justify="center" class="mt-5">
          <v-btn
            color="primary text--secondary"
            :disabled="!valid"
            @click="login"
            >Enviar</v-btn
          >
        </v-row>
      </v-card-text>

      <v-row justify="center" align="center" class="ma-0">
        <v-col sm="6" cols="12">
          <div class="mx-auto text-center">
            <i>Centro de informtica - CIn UFPE</i>
          </div>
        </v-col>
        <v-col sm="6" cols="12">
          <v-img
            class="mx-auto"
            src="../assets/cin_logo.png"
            contain
            height="100"
          />
        </v-col>
      </v-row>
    </v-card>
  </v-container>
</template>

<script>
import { non_empty_rule } from "../utils/form";
import { mapActions } from "vuex";

export default {
  data: () => ({
    valid: false,
    key: "",
    file: null,
  }),
  methods: {
    non_empty_rule: non_empty_rule,
    ...mapActions("inference", ["getInference"]),
    login() {
      if (this.$refs.form.validate()) {
        this.getInference({ key: this.key, file: this.file }).then(() => {
          this.$router.push("/home");
        });
      }
    },
  },
};
</script>
```

File: ./src/views/Home.vue

```
<template>
  <v-container fluid class="primary">
    <v-row class="my-10 px-5" align="center" justify="center">
      <v-img src="../assets/logo.png" contain max-width="50" max-height="50" />
      <div class="ml-5 text-h4 text-break text-center">DOUTOR INTELIGENTE</div>
    </v-row>

    <v-card class="mx-auto pa-10 rounded-tr-xl" max-width="700">
      <v-card v-for="(inference, i) in response" :key="i" class="my-5">
        <v-row align="center" justify="center" class="ma-0">
          <v-col sm="7" cols="12">
            <v-img :src="inference.heatmap_link" contain max-height="400" />
          </v-col>
          <v-col sm="5" cols="12">
            <v-row
              class="text-h5 font-weight-black text-break"
              justify="center"
            >
              {{ inference.pathology }}
            </v-row>
            <v-row justify="center" class="mb-5"
              >({{ (inference.prediction * 100).toFixed(2) }}%)</v-row
            >
          </v-col>
        </v-row>
      </v-card>
    </v-card>
    <v-btn
      elevation="2"
      fab
      fixed
      left
      bottom
      x-large
      @click="backConfirmation = true"
    >
      <v-icon>fas fa-arrow-left</v-icon>
    </v-btn>

    <v-btn
      elevation="2"
      fab
      fixed
      right
      bottom
      x-large
      @click="openEmailForm = true"
    >
      <v-icon>fas fa-envelope</v-icon>
    </v-btn>

    <v-dialog persistent v-model="backConfirmation" max-width="500">
      <ConfirmationBox
        message="Voltar para a tela inicial?"
        @yes="back"
        @no="backConfirmation = false"
      />
    </v-dialog>

    <v-dialog persistent v-model="openEmailForm" max-width="800">
      <MailForm @cancel="openEmailForm = false" @send="sendEmail" />
    </v-dialog>
  </v-container>
</template>

<script>
import { mapActions, mapState } from "vuex";
import ConfirmationBox from "../components/ConfirmationBox.vue";
import MailForm from "../components/MailForm.vue";

export default {
  data: () => ({
    inferences: [
      {
        pathology: "Massa",
        probability: 0.95,
      },
      {
        pathology: "Cardiomegalia",
        probability: 0.48,
      },
    ],
    backConfirmation: false,
    openEmailForm: false,
  }),
  computed: {
    ...mapState("inference", ["response"]),
  },
  components: {
    ConfirmationBox,
    MailForm,
  },
  methods: {
    ...mapActions("auth", ["signout"]),
    back() {
      this.signout().then(() => this.$router.push("/signin"));
    },
    sendEmail(email) {
      this.openEmailForm = false;

      console.log("SEND EMAIL", email);
    },
  },
};
</script>
```

File: ./src/views/About.vue

```
<template>
  <div class="about">
    <h1>This is an about page</h1>
  </div>
</template>
```

File: ./src/App.vue

```
<template>
  <v-app>
    <Alert />
    <Spinner />
    <router-view />
  </v-app>
</template>

<script>
import Spinner from "./components/Spinner.vue";
import Alert from "./components/Alert.vue";

export default {
  components: {
    Spinner,
    Alert,
  },
};
</script>
```

File: ./src/assets/cin_logo.png

[binary]

File: ./src/assets/heatmap.png

[binary]

File: ./src/assets/logo.png

[binary]

File: ./src/main.js

```
import Vue from "vue";
import App from "./App.vue";
import router from "./router";
import store from "./store";
import vuetify from "./plugins/vuetify";
import "@babel/polyfill";

Vue.config.productionTip = false;

new Vue({
  router,
  store,
  vuetify,
  render: (h) => h(App),
}).$mount("#app");
```

File: ./src/plugins/vuetify.js

```
import Vue from "vue";
import Vuetify from "vuetify";
import "vuetify/dist/vuetify.min.css";
import pt from "vuetify/lib/locale/pt";
import "@fortawesome/fontawesome-free/css/all.css";

Vue.use(Vuetify);

export default new Vuetify({
  lang: {
    locales: { pt },
    current: "pt",
  },
  icons: {
    iconfont: "fa",
  },
  theme: {
    themes: {
      light: {
        primary: "#BBDEFB",
        secondary: "#000000",
      },
    },
  },
});
```

File: ./src/store/inference.js

```
import session from "../api/session";

const initialState = {
  response: null,
};

const actions = {
  getInference({ commit }, { key, file }) {
    commit("LOADING", true, { root: true });

    var formData = new FormData();
    formData.append("originalImage", file);
    formData.append("key", key);

    return session
      .post("/inference/", formData)
      .then((response) => {
        commit("auth/SET_TOKEN", key, { root: true });

        response.data.sort((firstEl, secondEl) => {
          if (firstEl.prediction < secondEl.prediction) return 1;
          else return -1;
        });
        commit("SET_RESPONSE", response.data);
        commit("SET_TOKEN", key);

        commit("LOADING", false, { root: true });
      })
      .catch((error) => {
        commit("LOADING", false, { root: true });
        commit("ALERT", error.response.data.error, { root: true });
      });
  },
};

const mutations = {
  ["SET_RESPONSE"](state, response) {
    state.response = response;
  },
  ["REMOVE_RESPONSE"](state) {
    state.response = null;
  },
};

export default {
  namespaced: true,
  state: initialState,
  actions,
  mutations,
};
```

File: ./src/store/auth.js

```
const getters = { isAuthenticated: (state) => !!state.token };

const initialState = {
  token: null,
};

const actions = {
  login({ commit }, token) {
    commit("SET_TOKEN", token);
  },
  signout({ commit }) {
    commit("REMOVE_TOKEN");
  },
};

const mutations = {
  ["SET_TOKEN"](state, token) {
    state.token = token;
  },
  ["REMOVE_TOKEN"](state) {
    state.token = null;
  },
};

export default {
  namespaced: true,
  state: initialState,
  getters,
  actions,
  mutations,
};
```

File: ./src/store/index.js

```
import Vue from "vue";
import Vuex from "vuex";

import auth from "./auth";
import inference from "./inference";

Vue.use(Vuex);

export default new Vuex.Store({
  state: {
    loading: false,
    alertMessage: "",
  },
  mutations: {
    ["LOADING"](state, loadingState) {
      state.loading = loadingState;
    },
    ["ALERT"](state, message) {
      state.alertMessage = message;
      setTimeout(() => (state.alertMessage = ""), 100);
    },
  },
  actions: {},
  modules: {
    auth,
    inference,
  },
});
```

File: ./src/utils/form.js

```
export function non_empty_rule(value) {
  if (typeof value == "string") {
    if (value.length > 0) return true;
    else return "Campo Obrigatrio";
  } else {
    if (value?.size > 0) return true;
    else return "Campo Obrigatrio";
  }
}

export function email_rule(email) {
  if (email == "") return "Campo Obrigatrio";
  if (!/.+@.+\..+/.test(email)) return "E-mail invlido";
  return true;
}
```

File: ./src/components/HelloWorld.vue

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
    <p>
      For a guide and recipes on how to configure / customize this project,<br />
      check out the
      <a href="https://cli.vuejs.org" target="_blank" rel="noopener"
        >vue-cli documentation</a
      >.
    </p>
    <h3>Installed CLI Plugins</h3>
    <ul>
      <li>
        <a
          href="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-babel"
          target="_blank"
          rel="noopener"
          >babel</a
        >
      </li>
      <li>
        <a
          href="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-router"
          target="_blank"
          rel="noopener"
          >router</a
        >
      </li>
      <li>
        <a
          href="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-vuex"
          target="_blank"
          rel="noopener"
          >vuex</a
        >
      </li>
      <li>
        <a
          href="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-eslint"
          target="_blank"
          rel="noopener"
          >eslint</a
        >
      </li>
    </ul>
    <h3>Essential Links</h3>
    <ul>
      <li>
        <a href="https://vuejs.org" target="_blank" rel="noopener">Core Docs</a>
      </li>
      <li>
        <a href="https://forum.vuejs.org" target="_blank" rel="noopener"
          >Forum</a
        >
      </li>
      <li>
        <a href="https://chat.vuejs.org" target="_blank" rel="noopener"
          >Community Chat</a
        >
      </li>
      <li>
        <a href="https://twitter.com/vuejs" target="_blank" rel="noopener"
          >Twitter</a
        >
      </li>
      <li>
        <a href="https://news.vuejs.org" target="_blank" rel="noopener">News</a>
      </li>
    </ul>
    <h3>Ecosystem</h3>
    <ul>
      <li>
        <a href="https://router.vuejs.org" target="_blank" rel="noopener"
          >vue-router</a
        >
      </li>
      <li>
        <a href="https://vuex.vuejs.org" target="_blank" rel="noopener">vuex</a>
      </li>
      <li>
        <a
          href="https://github.com/vuejs/vue-devtools#vue-devtools"
          target="_blank"
          rel="noopener"
          >vue-devtools</a
        >
      </li>
      <li>
        <a href="https://vue-loader.vuejs.org" target="_blank" rel="noopener"
          >vue-loader</a
        >
      </li>
      <li>
        <a
          href="https://github.com/vuejs/awesome-vue"
          target="_blank"
          rel="noopener"
          >awesome-vue</a
        >
      </li>
    </ul>
  </div>
</template>

<script>
export default {
  name: "HelloWorld",
  props: {
    msg: String,
  },
};
</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped lang="scss">
h3 {
  margin: 40px 0 0;
}
ul {
  list-style-type: none;
  padding: 0;
}
li {
  display: inline-block;
  margin: 0 10px;
}
a {
  color: #42b983;
}
</style>
```

File: ./src/components/Alert.vue

```
<template>
  <v-snackbar v-model="alert" top color="error" timeout="5000" class="headline">
    <v-icon class="mr-5">fas fa-exclamation-triangle</v-icon>
    {{ message }}
  </v-snackbar>
</template>

<script>
import { mapState } from "vuex";

export default {
  data: () => ({
    alert: false,
    message: "",
  }),
  computed: {
    ...mapState(["alertMessage"]),
  },
  watch: {
    alertMessage() {
      this.alert = true;
      if (this.alertMessage != "") this.message = this.alertMessage;
    },
  },
};
</script>
```

File: ./src/components/ConfirmationBox.vue

```
<template>
  <v-card max-width="500">
    <v-card-title>{{ message }}</v-card-title>

    <v-card-actions>
      <v-spacer />
      <v-btn text @click="yes">Sim</v-btn>
      <v-btn text color="red" @click="no">No</v-btn>
    </v-card-actions>
  </v-card>
</template>

<script>
export default {
  props: {
    message: {
      type: String,
      default: "Voc confirma sua ao?",
    },
  },
  methods: {
    yes() {
      this.$emit("yes");
    },
    no() {
      this.$emit("no");
    },
  },
};
</script>
```

File: ./src/components/Spinner.vue

```
<template>
  <v-overlay :value="loading">
    <v-progress-circular indeterminate size="64"></v-progress-circular>
  </v-overlay>
</template>

<script>
import { mapState } from "vuex";

export default {
  computed: {
    ...mapState(["loading"]),
  },
};
</script>
```

File: ./src/components/MailForm.vue

```
<template>
  <v-card>
    <v-card-title>Enviar resultados por email</v-card-title>

    <v-card-text>
      <v-form
        ref="form"
        lazy-validation
        v-model="valid"
        @keyup.native.enter="send"
      >
        <v-text-field
          v-model="email"
          label="E-mail"
          :rules="[email_rule]"
          outlined
          clearable
          clear-icon="fas fa-times"
          color="secondary"
        ></v-text-field>
      </v-form>
    </v-card-text>

    <v-card-actions>
      <v-spacer />
      <v-btn color="primary text--secondary" @click="send" :disabled="!valid">
        Enviar
      </v-btn>
      <v-btn color="red" text @click="cancel">Cancelar</v-btn>
    </v-card-actions>
  </v-card>
</template>

<script>
import { email_rule } from "../utils/form";

export default {
  data: () => ({
    valid: false,
    email: "",
  }),
  methods: {
    email_rule: email_rule,
    cancel() {
      this.$emit("cancel");
    },
    send() {
      if (this.$refs.form.validate()) {
        this.$emit("send", this.email);
      }
    },
  },
};
</script>
```

File: ./src/router/index.js

```
import Vue from "vue";
import VueRouter from "vue-router";

import Home from "../views/Home.vue";
import SignIn from "../views/SignIn.vue";

import store from "../store";

Vue.use(VueRouter);

const requireAuthenticated = (to, from, next) => {
  if (!store.getters["auth/isAuthenticated"]) next("/signin");
  else next();
};

const requireUnauthenticated = (to, from, next) => {
  if (!store.getters["auth/isAuthenticated"]) next();
  else next("/home");
};

const routes = [
  {
    path: "/",
    redirect: "/home",
  },
  {
    path: "/home",
    name: "Home",
    component: Home,
    beforeEnter: requireAuthenticated,
  },
  {
    path: "/signin",
    name: "SignIn",
    component: SignIn,
    beforeEnter: requireUnauthenticated,
  },
];

const router = new VueRouter({
  routes,
});

export default router;
```

File: ./src/api/session.js

```
import axios from "axios";

const session = axios.create({
  //baseURL: "http://localhost:8000/",
  baseURL: "http://ec2-3-140-18-246.us-east-2.compute.amazonaws.com/",
  headers: {
    //Authorization: "Api-Key b8TnfSIq.uq8vcQqbCEERu263q6HNt8tdeyBeFOvs",
    //Authorization: "Api-Key Tq3LIX4L.7IIqbfnbq7gcAhR8jD8etBbmdkUbCpTr",
    Authorization: "Api-Key L5O5vCrq.goUtmiHaHQQcYo3HZFMRtSRpdtNLkiiz",
  },
});

export default session;
```

File: ./babel.config.js

```
module.exports = {
  presets: ["@vue/cli-plugin-babel/preset"],
};
```

File: ./.gitignore

```
.DS_Store
node_modules
/dist


# local env files
.env.local
.env.*.local

# Log files
npm-debug.log*
yarn-debug.log*
yarn-error.log*
pnpm-debug.log*

# Editor directories and files
.idea
.vscode
*.suo
*.ntvs*
*.njsproj
*.sln
*.sw?
```

File: ./package.json

```
[binary]
```

File: ./README.md

~~~
# doutor-inteligente-front-end

## Project setup
```
npm install
```

### Compiles and hot-reloads for development
```
npm run serve
```

### Compiles and minifies for production
```
npm run build
```

### Lints and fixes files
```
npm run lint
```

### Customize configuration
See [Configuration Reference](https://cli.vuejs.org/config/).
~~~

File: ./.eslintrc.js

```
module.exports = {
  root: true,
  env: {
    node: true,
  },
  extends: ["plugin:vue/essential", "eslint:recommended", "@vue/prettier"],
  parserOptions: {
    parser: "babel-eslint",
  },
  rules: {
    "no-console": process.env.NODE_ENV === "production" ? "warn" : "off",
    "no-debugger": process.env.NODE_ENV === "production" ? "warn" : "off",
    "prettier/prettier": [
      "error",
      {
        endOfLine: "auto",
      },
    ],
  },
};
```

File: ./.git/index

[binary]

File: ./.git/HEAD

```
ref: refs/heads/master
```

File: ./.git/packed-refs

```
# pack-refs with: peeled fully-peeled sorted
30e758957e988a49645b34cbb483d316e54acf61 refs/remotes/origin/master
```

File: ./.git/config

```
[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
[remote "origin"]
url = https://github.com/gabrielsm0405/doutor-digital-front-end.git
fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
remote = origin
merge = refs/heads/master
```

File: ./.git/description

```
Unnamed repository; edit this file 'description' to name the repository.
```

File: ./.git/hooks/fsmonitor-watchman.sample

```
#!/usr/bin/perl

use strict;
use warnings;
use IPC::Open2;

# An example hook script to integrate Watchman
# (https://facebook.github.io/watchman/) with git to speed up detecting
# new and modified files.
#
# The hook is passed a version (currently 1) and a time in nanoseconds
# formatted as a string and outputs to stdout all files that have been
# modified since the given time. Paths must be relative to the root of
# the working tree and separated by a single NUL.
```

```
#
# To enable this hook, rename this file to "query-watchman" and set
# 'git config core.fsmonitor .git/hooks/query-watchman'
#
my ($version, $time) = @ARGV;

# Check the hook interface version

if ($version == 1) {
# convert nanoseconds to seconds
# subtract one second to make sure watchman will return all changes
$time = int ($time / 1000000000) - 1;
} else {
die "Unsupported query-fsmonitor hook version '$version'.\n" .
"Falling back to scanning...\n";
}

my $git_work_tree;
if ($^O =~ 'msys' || $^O =~ 'cygwin') {
$git_work_tree = Win32::GetCwd();
$git_work_tree =~ tr/\\/\//;
} else {
require Cwd;
$git_work_tree = Cwd::cwd();
}

my $retry = 1;

launch_watchman();

sub launch_watchman {

my $pid = open2(\*CHLD_OUT, \*CHLD_IN, 'watchman -j --no-pretty')
or die "open2() failed: $!\n" .
"Falling back to scanning...\n";

# In the query expression below we're asking for names of files that
# changed since $time but were not transient (ie created after
# $time but no longer exist).
#
# To accomplish this, we're using the "since" generator to use the
# recency index to select candidate nodes and "fields" to limit the
# output to file names only.

my $query = <<" END";
["query", "$git_work_tree", {
"since": $time,
"fields": ["name"]
}]
END

print CHLD_IN $query;
close CHLD_IN;
my $response = do {local $/; <CHLD_OUT>};

die "Watchman: command returned no output.\n" .
"Falling back to scanning...\n" if $response eq "";
die "Watchman: command returned invalid output: $response\n" .
"Falling back to scanning...\n" unless $response =~ /^\{/;

my $json_pkg;
eval {
require JSON::XS;
$json_pkg = "JSON::XS";
1;
} or do {
require JSON::PP;
$json_pkg = "JSON::PP";
};

my $o = $json_pkg->new->utf8->decode($response);

if ($retry > 0 and $o->{error} and $o->{error} =~ m/unable to resolve root .* directory (.*) is not watched/) {
print STDERR "Adding '$git_work_tree' to watchman's watch list.\n";
$retry--;
qx/watchman watch "$git_work_tree"/;
die "Failed to make watchman watch '$git_work_tree'.\n" .
"Falling back to scanning...\n" if $? != 0;

# Watchman will always return all files on the first query so
# return the fast "everything is dirty" flag to git and do the
# Watchman query just to get it over with now so we won't pay
# the cost in git to look up each individual file.
print "/\0";
eval { launch_watchman() };
exit 0;
}

die "Watchman: $o->{error}.\n" .
"Falling back to scanning...\n" if $o->{error};

binmode STDOUT, ":utf8";
local $, = "\0";
print @{$o->{files}};
}
```

File: ./.git/hooks/update.sample

```
#!/bin/sh
#
# An example hook script to block unannotated tags from entering.
# Called by "git receive-pack" with arguments: refname sha1-old sha1-new
#
# To enable this hook, rename this file to "update".
#
# Config
# ------
# hooks.allowunannotated
# This boolean sets whether unannotated tags will be allowed into the
# repository. By default they won't be.
# hooks.allowdeletetag
# This boolean sets whether deleting tags will be allowed in the
# repository. By default they won't be.
# hooks.allowmodifytag
# This boolean sets whether a tag may be modified after creation. By default
# it won't be.
# hooks.allowdeletebranch
# This boolean sets whether deleting branches will be allowed in the
# repository. By default they won't be.
# hooks.denycreatebranch
# This boolean sets whether remotely creating branches will be denied
# in the repository. By default this is allowed.
#

# --- Command line
refname="$1"
oldrev="$2"
newrev="$3"

# --- Safety check
if [ -z "$GIT_DIR" ]; then
echo "Don't run this script from the command line." >&2
echo " (if you want, you could supply GIT_DIR then run" >&2
echo " $0 <ref> <oldrev> <newrev>)" >&2
exit 1
fi

if [ -z "$refname" -o -z "$oldrev" -o -z "$newrev" ]; then
echo "usage: $0 <ref> <oldrev> <newrev>" >&2
exit 1
fi

# --- Config
allowunannotated=$(git config --bool hooks.allowunannotated)
allowdeletebranch=$(git config --bool hooks.allowdeletebranch)
denycreatebranch=$(git config --bool hooks.denycreatebranch)
allowdeletetag=$(git config --bool hooks.allowdeletetag)
allowmodifytag=$(git config --bool hooks.allowmodifytag)

# check for no description
projectdesc=$(sed -e '1q' "$GIT_DIR/description")
case "$projectdesc" in
"Unnamed repository"* | "")
echo "*** Project description file hasn't been set" >&2
exit 1
;;
esac

# --- Check types
# if $newrev is 0000...0000, it's a commit to delete a ref.
zero="0000000000000000000000000000000000000000"
if [ "$newrev" = "$zero" ]; then
newrev_type=delete
else
newrev_type=$(git cat-file -t $newrev)
fi

case "$refname","$newrev_type" in
refs/tags/*,commit)
# un-annotated tag
short_refname=${refname##refs/tags/}
if [ "$allowunannotated" != "true" ]; then
echo "*** The un-annotated tag, $short_refname, is not allowed in this repository" >&2
echo "*** Use 'git tag [ -a | -s ]' for tags you want to propagate." >&2
exit 1
fi
;;
refs/tags/*,delete)
# delete tag
if [ "$allowdeletetag" != "true" ]; then
echo "*** Deleting a tag is not allowed in this repository" >&2
exit 1
fi
;;
refs/tags/*,tag)
# annotated tag
if [ "$allowmodifytag" != "true" ] && git rev-parse $refname > /dev/null 2>&1
then
echo "*** Tag '$refname' already exists." >&2
echo "*** Modifying a tag is not allowed in this repository." >&2
exit 1
fi
;;
refs/heads/*,commit)
# branch
if [ "$oldrev" = "$zero" -a "$denycreatebranch" = "true" ]; then
echo "*** Creating a branch is not allowed in this repository" >&2
exit 1
fi
;;
refs/heads/*,delete)
# delete branch
if [ "$allowdeletebranch" != "true" ]; then
echo "*** Deleting a branch is not allowed in this repository" >&2
```

```
exit 1
fi
;;
refs/remotes/*,commit)
# tracking branch
;;
refs/remotes/*,delete)
# delete tracking branch
if [ "$allowdeletebranch" != "true" ]; then
echo "*** Deleting a tracking branch is not allowed in this repository" >&2
exit 1
fi
;;
*)
# Anything else (is there anything else?)
echo "*** Update hook: unknown type of update to ref $refname of type $newrev_type" >&2
exit 1
;;
esac

# --- Finished
exit 0
```

File: ./.git/hooks/pre-applypatch.sample

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed
# by applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-applypatch".

. git-sh-setup
precommit="$(git rev-parse --git-path hooks/pre-commit)"
test -x "$precommit" && exec "$precommit" ${1+"$@"}
:
```

File: ./.git/hooks/pre-push.sample

```
#!/bin/sh

# An example hook script to verify what is about to be pushed. Called by "git
# push" after it has checked the remote status, but before anything has been
# pushed. If this script exits with a non-zero status nothing will be pushed.
#
# This hook is called with the following parameters:
#
# $1 -- Name of the remote to which the push is being done
# $2 -- URL to which the push is being done
#
# If pushing without using a named remote those arguments will be equal.
#
# Information about the commits which are being pushed is supplied as lines to
# the standard input in the form:
#
# <local ref> <local sha1> <remote ref> <remote sha1>
#
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

z40=0000000000000000000000000000000000000000

while read local_ref local_sha remote_ref remote_sha
do
if [ "$local_sha" = $z40 ]
then
# Handle delete
:
else
if [ "$remote_sha" = $z40 ]
then
# New branch, examine all commits
range="$local_sha"
else
# Update to existing branch, examine new commits
range="$remote_sha..$local_sha"
fi

# Check for WIP commit
commit=`git rev-list -n 1 --grep '^WIP' "$range"`
if [ -n "$commit" ]
then
echo >&2 "Found WIP commit in $local_ref, not pushing"
exit 1
fi
fi
done

exit 0
```

File: ./.git/hooks/pre-receive.sample

```
#!/bin/sh
#
# An example hook script to make use of push options.
# The example simply echoes all push options that start with 'echoback='
# and rejects all pushes when the "reject" push option is used.
#
# To enable this hook, rename this file to "pre-receive".

if test -n "$GIT_PUSH_OPTION_COUNT"
then
i=0
while test "$i" -lt "$GIT_PUSH_OPTION_COUNT"
do
eval "value=\$GIT_PUSH_OPTION_$i"
case "$value" in
echoback=*)
echo "echo from the pre-receive-hook: ${value#*=}" >&2
;;
reject)
exit 1
esac
i=$((i + 1))
done
fi
```

File: ./.git/hooks/pre-merge-commit.sample

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git merge" with no arguments. The hook should
# exit with non-zero status after issuing an appropriate message to
# stderr if it wants to stop the merge commit.
#
# To enable this hook, rename this file to "pre-merge-commit".

. git-sh-setup
test -x "$GIT_DIR/hooks/pre-commit" &&
exec "$GIT_DIR/hooks/pre-commit"
:
```

File: ./.git/hooks/applypatch-msg.sample

```
#!/bin/sh
#
# An example hook script to check the commit log message taken by
# applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit. The hook is
# allowed to edit the commit message file.
#
# To enable this hook, rename this file to "applypatch-msg".

. git-sh-setup
commitmsg="$(git rev-parse --git-path hooks/commit-msg)"
test -x "$commitmsg" && exec "$commitmsg" ${1+"$@"}
:
```

File: ./.git/hooks/pre-commit.sample

```
#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments. The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

if git rev-parse --verify HEAD >/dev/null 2>&1
then
against=HEAD
else
# Initial commit: diff against an empty tree object
against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2

# Cross platform projects tend to avoid non-ASCII filenames; prevent
# them from being added to the repository. We exploit the fact that the
# printable range starts at the space character and ends with tilde.
if [ "$allownonascii" != "true" ] &&
# Note that the use of brackets around a tr range is ok here, (it's
# even required, for portability to Solaris 10's /usr/bin/tr), since
# the square bracket bytes happen to fall in the designated range.
test $(git diff --cached --name-only --diff-filter=A -z $against |
LC_ALL=C tr -d '[ -~]\0' | wc -c) != 0
then
cat <<\EOF
Error: Attempt to add a non-ASCII file name.

This can cause problems if you want to work with people on other platforms.
```

```
To be portable it is advisable to rename the file.

If you know what you are doing you can disable this check using:

git config hooks.allownonascii true
EOF
exit 1
fi

# If there are whitespace errors, print the offending file names and fail.
exec git diff-index --check --cached $against --
```

File: ./.git/hooks/prepare-commit-msg.sample

```
#!/bin/sh
#
# An example hook script to prepare the commit log message.
# Called by "git commit" with the name of the file that has the
# commit message, followed by the description of the commit
# message's source. The hook's purpose is to edit the commit
# message file. If the hook fails with a non-zero status,
# the commit is aborted.
#
# To enable this hook, rename this file to "prepare-commit-msg".

# This hook includes three examples. The first one removes the
# "# Please enter the commit message..." help message.
#
# The second includes the output of "git diff --name-status -r"
# into the message, just before the "git status" output. It is
# commented because it doesn't cope with --amend or with squashed
# commits.
#
# The third example adds a Signed-off-by line to the message, that can
# still be edited. This is rarely a good idea.

COMMIT_MSG_FILE=$1
COMMIT_SOURCE=$2
SHA1=$3

/usr/bin/perl -i.bak -ne 'print unless(m/^. Please enter the commit message/..m/^#$/)' "$COMMIT_MSG_FILE"

# case "$COMMIT_SOURCE,$SHA1" in
# ,|template,)
# /usr/bin/perl -i.bak -pe '
# print "\n" . `git diff --cached --name-status -r`
# if /^#/ && $first++ == 0' "$COMMIT_MSG_FILE" ;;
# *) ;;
# esac

# SOB=$(git var GIT_COMMITTER_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# git interpret-trailers --in-place --trailer "$SOB" "$COMMIT_MSG_FILE"
# if test -z "$COMMIT_SOURCE"
# then
# /usr/bin/perl -i.bak -pe 'print "\n" if !$first_line++' "$COMMIT_MSG_FILE"
# fi
```

File: ./.git/hooks/commit-msg.sample

```
#!/bin/sh
#
# An example hook script to check the commit log message.
# Called by "git commit" with one argument, the name of the file
# that has the commit message. The hook should exit with non-zero
# status after issuing an appropriate message if it wants to stop the
# commit. The hook is allowed to edit the commit message file.
#
# To enable this hook, rename this file to "commit-msg".

# Uncomment the below to add a Signed-off-by line to the message.
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg
# hook is more suited to it.
#
# SOB=$(git var GIT_AUTHOR_IDENT | sed -n 's/^\(.*>\).*$/Signed-off-by: \1/p')
# grep -qs "^$SOB" "$1" || echo "$SOB" >> "$1"

# This example catches duplicate Signed-off-by lines.

test "" = "$(grep '^Signed-off-by: ' "$1" |
sort | uniq -c | sed -e '/^[ ]*1[ ]/d')" || {
echo >&2 Duplicate Signed-off-by lines.
exit 1
}
```

File: ./.git/hooks/post-update.sample

```
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, rename this file to "post-update".

exec git update-server-info
```

File: ./.git/hooks/pre-rebase.sample

```
#!/bin/sh
#
# Copyright (c) 2006, 2008 Junio C Hamano
#
# The "pre-rebase" hook is run just before "git rebase" starts doing
# its job, and can prevent the command from running by exiting with
# non-zero status.
#
# The hook is called with the following parameters:
#
# $1 -- the upstream the series was forked from.
# $2 -- the branch being rebased (or empty when rebasing the current branch).
#
# This sample shows how to prevent topic branches that are already
# merged to 'next' branch from getting rebased, because allowing it
# would result in rebasing already published history.

publish=next
basebranch="$1"
if test "$#" = 2
then
topic="refs/heads/$2"
else
topic=`git symbolic-ref HEAD` ||
exit 0 ;# we do not interrupt rebasing detached HEAD
fi

case "$topic" in
refs/heads/??/*)
;;
*)
exit 0 ;# we do not interrupt others.
;;
esac

# Now we are dealing with a topic branch being rebased
# on top of master. Is it OK to rebase it?

# Does the topic really exist?
git show-ref -q "$topic" || {
echo >&2 "No such branch $topic"
exit 1
}

# Is topic fully merged to master?
not_in_master=`git rev-list --pretty=oneline ^master "$topic"`
if test -z "$not_in_master"
then
echo >&2 "$topic is fully merged to master; better remove it."
exit 1 ;# we could allow it, but there is no point.
fi

# Is topic ever merged to next? If so you should not be rebasing it.
only_next_1=`git rev-list ^master "^$topic" ${publish} | sort`
only_next_2=`git rev-list ^master ${publish} | sort`
if test "$only_next_1" = "$only_next_2"
then
not_in_topic=`git rev-list "^$topic" master`
if test -z "$not_in_topic"
then
echo >&2 "$topic is already up to date with master"
exit 1 ;# we could allow it, but there is no point.
else
exit 0
fi
else
not_in_next=`git rev-list --pretty=oneline ^${publish} "$topic"`
/usr/bin/perl -e '
my $topic = $ARGV[0];
my $msg = "* $topic has commits already merged to public branch:\n";
my (%not_in_next) = map {
/^([0-9a-f]+) /;
($1 => 1);
} split(/\n/, $ARGV[1]);
for my $elem (map {
/^([0-9a-f]+) (.*)$/;
[$1 => $2];
} split(/\n/, $ARGV[2])) {
if (!exists $not_in_next{$elem->[0]}) {
if ($msg) {
print STDERR $msg;
undef $msg;
}
print STDERR " $elem->[1]\n";
}
}
' "$topic" "$not_in_next" "$not_in_master"
exit 1
fi

<<\DOC_END

This sample hook safeguards topic branches that have been
published from being rewound.

The workflow assumed here is:

* Once a topic branch forks from "master", "master" is never
merged into it again (either directly or indirectly).

* Once a topic branch is fully cooked and merged into "master",
```

it is deleted. If you need to build on top of it to correct
earlier mistakes, a new topic branch is created by forking at
the tip of the "master". This is not strictly necessary, but
it makes it easier to keep your history simple.

* Whenever you need to test or publish your changes to topic
branches, merge them into "next" branch.

The script, being an example, hardcodes the publish branch name
to be "next", but it is trivial to make it configurable via
$GIT_DIR/config mechanism.

With this workflow, you would want to know:

(1) ... if a topic branch has ever been merged to "next". Young
topic branches can have stupid mistakes you would rather
clean up before publishing, and things that have not been
merged into other branches can be easily rebased without
affecting other people. But once it is published, you would
not want to rewind it.

(2) ... if a topic branch has been fully merged to "master".
Then you can delete it. More importantly, you should not
build on top of it -- other people may already want to
change things related to the topic as patches against your
"master", so if you need further changes, it is better to
fork the topic (perhaps with the same name) afresh from the
tip of "master".

Let's look at this example:

```
o---o---o---o---o---o---o---o---o---o "next"
/ / / /
/ a---a---b A / /
/ / / /
/ / c---c---c---c B /
/ / / \ /
/ / / b---b C \ /
/ / / / \ /
---o---o---o---o---o---o---o---o---o---o---o "master"
```

A, B and C are topic branches.

* A has one fix since it was merged up to "next".

* B has finished. It has been fully merged up to "master" and "next",
and is ready to be deleted.

* C has not merged to "next" at all.

We would want to allow C to be rebased, refuse A, and encourage
B to be deleted.

To compute (1):

```
git rev-list ^master ^topic next
git rev-list ^master next
```

if these match, topic has not merged in next at all.

To compute (2):

```
git rev-list master..topic
```

if this is empty, it is fully merged to "master".

DOC_END

File: ./.git/objects/6d/616f6471ba9ef676f1681013cdfb16ed7c4718

[binary]

File: ./.git/objects/0a/f497d718994dcc5ca338920236312dee8b7ba1

[binary]

File: ./.git/objects/8e/d00548770d3e14d8c913d71f359916cf826820

[binary]

File: ./.git/objects/06/1dad94948f5a8e5228517f50d3cfb41702eb3a

[binary]

File: ./.git/objects/96/1d267523dfac6b4e309fb419d8a4b8de2d8a42

[binary]

File: ./.git/objects/ad/e16ad8bf31fba43cab5fab5d17fca7049593d5

[binary]

File: ./.git/objects/a1/9f9ce8a228329cd911d3df38d29c64d82726a2

[binary]

File: ./.git/objects/53/9979e37ef5bb724f1b7a337af38e0616735653

[binary]

File: ./.git/objects/53/b231209e632ed947e8bbd2e508a009df16f5d9

[binary]

File: ./.git/objects/d7/193cc84ee2bd722ab7c422247c71819fa912be

[binary]

File: ./.git/objects/d7/85104deaf6b766addf8f82b4733f8ed6bd798c

[binary]

File: ./.git/objects/30/e758957e988a49645b34cbb483d316e54acf61

[binary]

File: ./.git/objects/73/ca04d985a2b5060d99285c186fa266ac372509

[binary]

File: ./.git/objects/5f/0fb0727fec5ca318ba98512fc76e1ca9c0d3e4

[binary]

File: ./.git/objects/f7/ed83c86de20fbc8ad5a34d1814eba5244e0c4f

[binary]

File: ./.git/objects/f7/482ed2b0575bd2d56d7c62429c4d380cbb73bf

[binary]

File: ./.git/objects/a8/d00ed60ae42bcd215d1af853b74521da82f651

[binary]

File: ./.git/objects/66/1cfd82e28c8fde171a7d4679928174a708c2f2

[binary]

File: ./.git/objects/fb/5dfbdb3138f7b900cf941cfec87db3fbbe9a41

[binary]

File: ./.git/objects/86/1600aa169b4d2ebc5ce85b948a9a14e0b2f3b1

[binary]

File: ./.git/objects/35/5f3ddb41a0836c97da43f380fa938bc1f5902d

[binary]

File: ./.git/objects/8d/4b8c35eead32d01a20fb9f97cc7269a3f9f9ab

[binary]

File: ./.git/objects/df/36fcfb72584e00488330b560ebcf34a41c64c2

[binary]

File: ./.git/objects/8b/5ac0fa3404a3dd2bb45545d611160d734d46f3

[binary]

File: ./.git/objects/98/2fcba9c7bc00235cbf7b0a6b96184dd4becd9b

[binary]

File: ./.git/objects/1e/6fe312bcc53145e02d30931424b0718349c966

[binary]

File: ./.git/objects/6a/4b86f1e7ebfb95f8de53b4e20d735a31b87297

[binary]

File: ./.git/objects/14/55026e7b448714bde4d1934cdcc66530c7bab3

[binary]

File: ./.git/objects/89/aca554563cdec198a0f11f186d1b2a2804de08

[binary]

File: ./.git/objects/37/f94b8b27461afc96ed54315d1764b666eb5080

[binary]

File: ./.git/objects/5c/08c4cfe444f287308ac278ba61f028c2b19762

[binary]

File: ./.git/objects/9b/7802e5b511377cf51676ee52c628dccabf9b3f

[binary]

File: ./.git/objects/61/cbd44ca9b52f26b7af949388145ee845b23806

[binary]

File: ./.git/objects/ff/6ee348b7757797924a550f754564784db925b7

[binary]

File: ./.git/objects/ff/754d4a212d8eb22cd216b078072203c29277e5

[binary]

File: ./.git/objects/40/3adbc1e527906a4aa59558cd582c20bcd1d738

[binary]

File: ./.git/objects/00/813ac0254b38b81fe7fe9db589afea02043c2c

[binary]

File: ./.git/objects/00/72c8de75fd86f9fac64290d39dfa29554557d4

[binary]

File: ./.git/objects/2e/6b1235e2bdd4850e035d97a7bab459c9e6061b

[binary]

File: ./.git/objects/4f/21c35466e126651d3723eef1eb06b2afd592de

[binary]

File: ./.git/objects/4f/ef6ff10e7f6a781e0513ed534c9f93dde8599a

[binary]

File: ./.git/objects/e1/e35fb9e9120d5a62eb9a666f9145ac1fad6679

[binary]

File: ./.git/objects/76/1bf36d14b389184ac8815457492b82586470c4

[binary]

File: ./.git/objects/05/709288991d1323485a556f62fb375434afe378

[binary]

File: ./.git/objects/4e/8bfd2811ef6b982c00fbe2eb100a3de76e5f44

[binary]

File: ./.git/objects/1f/14d89850078d32b2d35e8a2a131072f9e74191

[binary]

File: ./.git/objects/2d/ca4fcbdf6c6b5262ab4ec1beb2f92ecc62d096

[binary]

File: ./.git/objects/9a/3026d3e08e7d72db3e72d915eea6a4182a9816

[binary]

File: ./.git/objects/d1/97edfe2dab48f24f3bc500731272279dc665a4

[binary]

File: ./.git/objects/93/e88242d482c1761cf7b0bfa8afbd2347a3be25

[binary]

File: ./.git/objects/ef/18a06a2a2091a2fb7648a073b842531022282c

[binary]

File: ./.git/objects/70/dbddc122340d0701a27efd6e370bfb60256f6d

[binary]

File: ./.git/objects/f9/c6b4c165dc7ae7d07b87b3a7ccb2b2ec11399a

[binary]

File: ./.git/objects/92/fe466163ea4d6f809c7f1ddb8101578b379efc

[binary]

File: ./.git/objects/3f/a28070de24f2055171ca2e20543881cb7fdf1c

[binary]

File: ./.git/objects/c6/dc7418be0a86bcb3e1b687692975db0a05c126

[binary]

File: ./.git/objects/16/2a3ea97c29233aa898fa74630fde091e91680c

[binary]

File: ./.git/objects/f3/d2503fc2a44b5053b0837ebea6e87a2d339a43

[binary]

File: ./.git/objects/3a/9f57569130bf5956e6374448f4ef2686dc7cb7

[binary]

File: ./.git/objects/03/7237a2f3322031b68ecd5f42e88f0943487d52

[binary]

File: ./.git/objects/72/745a8d8cb042136b8f96b7139138b13f1a9c8d

[binary]

File: ./.git/objects/ea/a3035f06eb3c97d8ec4beeda5812017113b668

[binary]

File: ./.git/objects/90/fc2f15a1f1879bde076ba04edf7f0a9c161015

[binary]

File: ./.git/objects/1d/838472fb3a0ed1d863b588d60e5cef9732ab8b

[binary]

File: ./.git/objects/08/8342f6555719bbcd9c8830116804480f054da6

[binary]

File: ./.git/objects/13/3ef2df8c0f7d238c7fe67db6aa8d88c99d7fe1

[binary]

File: ./.git/objects/13/dedef9cc54297586288e2be095f8f1622c6e51

[binary]

File: ./.git/objects/bf/238bbd34318a6266b2b83917e8bf5810dafe08

[binary]

File: ./.git/objects/99/3404451281fa7976694ad3426b33470bdcf391

[binary]

File: ./.git/objects/ab/cb4cf1678b6522785c08491e7e7118a383bf9c

[binary]

File: ./.git/objects/f4/ff9e682fcffaf13df5ea79bba2cf07f7ca00ee

[binary]

File: ./.git/objects/fa/ec3d72719caaced7125734c121363cc4b9a387

[binary]

File: ./.git/objects/fa/115488736c5e0856adc0b448aa6e0d1e9f1c25

[binary]

File: ./.git/objects/1a/b7a295dd976871022f354bfb0a287034972e30

[binary]

File: ./.git/objects/5e/ffae2b4ecd7a1d6a34df1faf33573fa7855169

[binary]

File: ./.git/objects/eb/4bf4514ab2edabf322a5b07122ffe6fd5aeb29

[binary]

File: ./.git/objects/84/67b2f834ea3fc6b1a3b2c6a6b11206865d9b0d

[binary]

File: ./.git/objects/0d/49d17f0e671b61821276d3160ca8329aece4b7

[binary]

File: ./.git/objects/97/26c1b92b7b28600e63232df6232e1a6cd4296d

[binary]

File: ./.git/objects/65/e95d203e96ef1a91b9bfb37d446d4d35964884

[binary]

File: ./.git/objects/54/d4bed1ff4d78d8f8941a186d92172067c87b47

[binary]

File: ./.git/objects/38/16df418dcf39e2365f6fbd3a2149b938e122fc

[binary]

File: ./.git/objects/6e/3073f7ebc8bf186eb76b1fc139b2d51af02d01

[binary]

File: ./.git/info/exclude

```
# git ls-files --others --exclude-from=.git/info/exclude
# Lines that start with '#' are comments.
# For a project mostly in C, the following would be a good set of
# exclude patterns (uncomment them if you want to use them):
# *.[oa]
# *~
```

File: ./.git/refs/remotes/origin/HEAD

```
ref: refs/remotes/origin/master
```

File: ./.git/refs/heads/master

```
30e758957e988a49645b34cbb483d316e54acf61
```

File: ./.git/logs/HEAD

```
0000000000000000000000000000000000000000 30e758957e988a49645b34cbb483d316e54acf61 gabriel <gabriel@DESKTOP-MM4LSH9.localdomain>
1644590752 -0300 clone: from https://github.com/gabrielsm0405/doutor-digital-front-end.git
```

File: ./.git/logs/refs/remotes/origin/HEAD

```
0000000000000000000000000000000000000000 30e758957e988a49645b34cbb483d316e54acf61 gabriel <gabriel@DESKTOP-MM4LSH9.localdomain>
1644590752 -0300 clone: from https://github.com/gabrielsm0405/doutor-digital-front-end.git
```

File: ./.git/logs/refs/heads/master

```
0000000000000000000000000000000000000000 30e758957e988a49645b34cbb483d316e54acf61 gabriel <gabriel@DESKTOP-MM4LSH9.localdomain>
1644590752 -0300 clone: from https://github.com/gabrielsm0405/doutor-digital-front-end.git
```

File: ./public/favicon.ico

[binary]

File: ./public/index.html

```html
<!DOCTYPE html>
<html lang="pt">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title>Doutor Inteligente</title>
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/all.css" integrity="sha384-hWVjflwFxL6sNzntih27bfxkr27PmbbK/iSvJ+a4+0owXq79v+lsFkW54bOGbiDQ" crosso
  </head>
  <body>
    <noscript>
      <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

File: ./.browserslistrc

```
> 1%
last 2 versions
not dead

not ie <= 10
```