Models --2

admin:.....2

announcement:.....2

event:.....2

course:.....3

contact: 3

Views --4

add-admin:.....4

validate-admin:.....4

get-all-admins: 5

add-announcement:.....5

get-announcement-by-id: 6

get-all-announcements: 6

edit-announcement-by-id:.....7

remove-announcement-by-id:.....7

add-event: 8

get-event-by-id:.....8

get-all-events:.....9

edit-event-by-id:.....9

remove-event-by-id:.....10

add-course:.....10

get-course-by-id: 11

get-all-courses: 11

edit-course-by-id:.....12

remove-course-by-id:.....12

add-contact:.....13

get-contact-by-id: 13

get-all-contacts: 14

edit-contact-by-id: 14

remove-contact-by-id: 15

Southeast CS API v1.0.0

Models

File that defines the models of the database: ../southeastcsapi/api/models.py.

admin:

The **admin** model holds the information (username and password) for the users capable of using the Southeast CS Admin Panel. The following JSON string defines an **admin** object:

```
{
    "id": auto-created int,
    "username": str,
    "passwd": hashed str
}
```

announcement:

The **announcement** model holds the information (author, authored date, subject, and description) for general updates/news of the CS department. The following JSON string defines an **announcement** object:

```
{
    "id": auto-created int,
    "author": str,
    "authored_date": datetime as a str in format yyyy-MM-ddTHH:mm:tt,
    "subject": str,
    "description": str
}
```

event:

The **event** model holds the information (date of event, event name, location of event, description, and the organization putting on the event) for activities occurring in the CS department. The following JSON string defines an **event** object:

```
{
    "id": auto-created int,
    "date": datetime as a str in format yyyy-MM-ddTHH:mm:tt,
    "name": str,
    "location": str,
    "description": str,
    "organization": str
}
```

course:

The **course** model holds the information for all courses in the CS/CIS/CY majors.
The following JSON string defines an **course** object:

```
{
    "id": auto-created int,
    "name": str,
    "number": str,
    "credits": int,
    "prerequisites": str,
    "availability": str,
    "description": str
}
```

contact:

The **contact** model holds the information (name of contact, email, phone number, and office location) for helpful CS contacts, like: professors, IT help, student financial services, etc. The following JSON string defines a **contact** object:

```
{
    "id": auto-created int,
    "name" str,
    "email": str,
    "phone": str that must be length 10,
    "office": str
}
```

Views

The file that defines the views (or endpoints) of the API: ../southeastcsapi/api/views.py.

add-admin:

This view creates a new **admin** record in the database.

URL Path: http://127.0.0.1:8000/add-admin/

HTTP Method: POST

Returns: HTTP Status

Parameters: JSON string in the form

```
{
    "username": "admin",
    "passwd": "hashed password"
}
```

Troubleshooting:

201 - New admin created.
302 - Admin account already exists.
400 - Must use POST method or object was not in a valid form.

validate-admin:

This view validates an **admin's** login attempt.

URL Path: http://127.0.0.1:8000/validate-admin/

HTTP Method: POST

Returns: HTTP Status

Parameters: JSON string in the form

```
{
    "username": "admin",
    "passwd": "hashed password"
}
```

Troubleshooting:

200 - Login successful.
400 - Must use POST method or object was not in a valid form.
404 - Admin doesn't exist or password is incorrect.

get-all-admins:

This view returns a list of all the **admins** in the database.

URL Path: http://127.0.0.1:8000/get-all-admins/

HTTP Method: GET

Returns: HTTP Status or JSON string in the form of

```
[
    {
        "id": 1,
        "username": "admin",
        "passwd": "hashed password"
    },
    {
        "id": 2,
        "username": "admin2",
        "passwd": "hashed password"
    }
]
```

Parameters: N/A

Troubleshooting:

400 - Must use GET method.

add-announcement:

This view creates a new **announcement** record in the database.

URL Path: http://127.0.0.1:8000/add-announcement/

HTTP Method: POST

Returns: HTTP Status

Parameters: JSON string in the form of

```
{
    "author": "Sion Pixley",
    "authored_date": "2020-02-29T13:00:00",
    "subject": "New Computer Hardware in DH023",
    "description": "The CS department has purchased all new PCs for the CS/CY students."
}
```

Troubleshooting:

201 - New announcement created.
302 - Announcement already exists.
400 - Must use POST method or object was not in a valid form.

get-announcement-by-id:

This view returns a specific **announcement** for a given id.

URL Path: http://127.0.0.1:8000/get-announcement-by-id/id/
HTTP Method: GET
Returns: HTTP Status or JSON string in the form of

```
{
    "id": 1,
    "author": "Sion Pixley",
    "authored_date": "2020-02-29T13:00:00",
    "subject": "New Computer Hardware in DH023",
    "description": "The CS department has purchased all new PCs for the CS/CY students."
}
```

Parameters:
id - int
Troubleshooting:
400 - Must use GET method.
404 - Announcement does not exist.

get-all-announcements:

This view returns a list of all the **announcements** in the database

URL Path: http://127.0.0.1:8000/get-all-announcements/
HTTP Method: GET
Returns: HTTP Status or JSON string in the form of

```
[
    {
        "id": 1,
        "author": "Sion Pixley",
        "authored_date": "2020-02-29T13:00:00",
        "subject": "New computer hardware in DH023",
        "description": "The CS department has purchased all new PCs for..."
    },
    {
        "id": 2,
        "author": "Jeffrey Smith",
        "authored_date": "2020-02-29T14:20:00",
        "subject": "Reminder: Tomorrow is the last day to pay",
        "description": "Final payment for the CS field trip is due tomorrow at 2pm"
    }
]
```

Parameters: N/A
Troubleshooting:
400 - Must use GET method.

edit-announcement-by-id:

This view updates a specific **announcement**.

URL Path: http://127.0.0.1:8000/edit-announcement-by-id/id/

HTTP Method: POST

Returns: HTTP Status

Parameters:

id - int

AND a JSON string in the form of

```
{
    "author": "Jeffrey Smith",
    "authored_date": "2020-02-29T14:20:00",
    "subject": "Reminder: Tomorrow is the last day to pay",
    "description": "Final payment for the CS field trip is due tomorrow at 2pm"
}
```

Troubleshooting:

200 - Announcement updated.
400 - Must use POST method or object was not in a valid form.
404 - Announcement does not exist.

remove-announcement-by-id:

This view deletes an **announcement** from the database.

URL Path: http://127.0.0.1:8000/remove-announcement-by-id/id/

HTTP Method: DELETE

Returns: HTTP Status

Parameters:

id - int

Troubleshooting:

200 - Announcement removed from database.
400 - Must use DELETE method.
404 - Announcement does not exist.

add-event:

This view creates a new **event** record in the database.

URL Path: http://127.0.0.1:8000/add-event/

HTTP Method: POST

Returns: HTTP Status

Parameters: JSON string in the form of

```
{
    "date": "2020-02-29T13:00:00",
    "location": "Dempster 003",
    "name": "Pizza Party",
    "description": "Free pizza in Dempster 023 at 1pm. A presentation will be given by...",
    "organization": "CS Club"
}
```

Troubleshooting:

201 - New event created.
302 - Event already exists.
400 - Must use POST method or object was not in a valid form.

get-event-by-id:

This view returns a specific **event** record.

URL Path: http://127.0.0.1:8000/get-event-by-id/id/

HTTP Method: GET

Returns: HTTP Status or JSON string in the form of

```
{
    "id": 1,
    "date": "2020-02-29T13:00:00",
    "location": "Dempster 003",
    "name": "Pizza Party",
    "description": "Free pizza in Dempster 023 at 1pm. A presentation will be given by...",
    "organization": "CS Club"
}
```

Parameters:

id - int

Troubleshooting:

400 - Must use GET method.
404 - Event does not exist.

get-all-events:

This view returns a list of all the **events** in the database.

URL Path: http://127.0.0.1:8000/get-all-events/

HTTP Method: GET

Returns: JSON string in the form of

```
[
    {
        "id": 1,
        "date": "2020-02-29T13:00:00",
        "location": "Dempster 003",
        "name": "Pizza Party",
        "description": "Free pizza in Dempster 023 at 1pm. A presentation will be given by...",
        "organization": "CS Club"
    }
]
```

Parameters: N/A

Troubleshooting:

400 - Must use GET method.

edit-event-by-id:

This view updates an **event** record in the database.

URL Path: http://127.0.0.1:8000/edit-event-by-id/id/

HTTP Method: POST

Returns: HTTP Status

Parameters:

id - int

AND JSON string in the form of

```
{
    "date": "2020-02-29T13:00:00",
    "location": "Dempster 003",
    "name": "Pizza Party",
    "description": "Free pizza in Dempster 023 at 1pm. A presentation will be given by...",
    "organization": "CS Club"
}
```

Troubleshooting:

200 - Event updated.

400 - Must use POST method or object was not in a valid form.

404 - Event does not exist.

remove-event-by-id:

This view deletes an **event** from the database.

URL Path: http://127.0.0.1:8000/remove-event-by-id/id/
HTTP Method: DELETE
Returns: HTTP Status
Parameters:
id - int
Troubleshooting:
200 - Event removed from database.
400 - Must use DELETE method.
404 - Event does not exist.

add-course:

This view creates a new **course** record in the database.

URL Path: http://127.0.0.1:8000/add-course/
HTTP Method: POST
Returns: HTTP Status
Parameters: JSON string in the form of

```
{
    "name": "Intro to CS",
    "number": "CS100",
    "credits": 3,
    "prerequisites": "n/a",
    "availability": "Every spring and fall.",
    "description": "Entry-level CS course."
}
```

Troubleshooting:
201 - New course created.
302 - Course already exists.
400 - Must use POST method or object was not in a valid form.

get-course-by-id:

This view gets a specific **course** from the database.

URL Path: http://127.0.0.1:8000/get-course-by-id/id/

HTTP Method: GET

Returns: HTTP Status or JSON string in the form of

```
{
    "id": 1,
    "name": "Intro to CS",
    "number": "CS100",
    "credits": 3,
    "prerequisites": "n/a",
    "availability": "Every spring and fall.",
    "description": "Entry-level CS course."
}
```

Parameters:

id - int

Troubleshooting:

400 - Must use GET method.
404 - Course does not exist.

get-all-courses:

This view returns a list of all the **courses** in the database.

URL Path: http://127.0.0.1:8000/get-all-courses/

HTTP Method: GET

Returns: JSON string in the form of

```
[
    {
        "id": 1,
        "name": "Intro to CS",
        "number": "CS100",
        "credits": 3,
        "prerequisites": "n/a",
        "availability": "Every spring and fall.",
        "description": "Entry-level CS course."
    }
]
```

Parameters: N/A

Troubleshooting:

400 - Must use GET method.

edit-course-by-id:

This view updates a specific **course** in the database.

URL Path: http://127.0.0.1:8000/edit-course-by-id/id/

HTTP Method: POST

Returns: HTTP Status

Parameters:

id - int

Troubleshooting:

200 - Course updated.

400 - Must use POST method or object was not in a valid form.

404 - Course does not exist.

remove-course-by-id:

This view deletes a **course** from the database.

URL Path: http://127.0.0.1:8000/remove-course-by-id/id/

HTTP Method: DELETE

Returns: HTTP Status

Parameters:

id - int

Troubleshooting:

200 - Course removed from database.

400 - Must use DELETE method.

404 - Course does not exist.

add-contact:

This view creates a new **contact** in the database.

URL Path: http://127.0.0.1:8000/add-contact/

HTTP Method: POST

Returns: HTTP Status

Parameters: JSON string in the form of

```
{
    "name": "Sion Pixley",
    "email": "test@gmail.com",
    "phone": "5735557812",
    "office": "DH026"
}
```

Troubleshooting:

201 - New contact created.

302 - Contact already exists.

400 - Must use POST method or object was not in a valid form.

get-contact-by-id:

This view returns a specific **contact**.

URL Path: http://127.0.0.1:8000/get-contact-by-id/id/

HTTP Method: GET

Returns: HTTP Status or JSON string in the form of

```
{
    "id": 1,
    "name": "Sion Pixley",
    "email": "test@gmail.com",
    "phone": "5735557812",
    "office": "DH026"
}
```

Parameters:

id - int

Troubleshooting:

400 - Must use GET method.

404 - Contact does not exist.

get-all-contacts:

This view returns a list of all of the **contacts** in the database.

URL Path: http://127.0.0.1:8000/get-all-contacts/

HTTP Method: GET

Returns: HTTP Status or JSON string in the form of

```
[
    {
        "id": 1,
        "name": "Sion Pixley",
        "email": "test@gmail.com",
        "phone": "5735557812",
        "office": "DH026"
    }
]
```

Parameters: N/A

Troubleshooting:

400 - Must use GET method.

edit-contact-by-id:

This view edits a specific **contact** in the database.

URL Path: http://127.0.0.1:8000/edit-contact-by-id/id/

HTTP Method: POST

Returns: HTTP Status

Parameters:

id - int

AND JSON string in the form of

```
{
    "name": "Sion Pixley",
    "email": "test@gmail.com",
    "phone": "5735557812",
    "office": "DH026"
}
```

Troubleshooting:

200 - Contact updated.

400 - Must use POST method or object was not in a valid form.

404 - Contact does not exist.

remove-contact-by-id:

This view deletes a **contact** from the database.

URL Path: http://127.0.0.1:8000/remove-contact-by-id/id/

HTTP Method: DELETE

Returns: HTTP Status

Parameters:

id - int

Troubleshooting:

200 - Article removed from database.
400 - Must use DELETE method or object was not in a valid form.
404 - Contact does not exist.