Southeast CS API v1.0.0

<u>Models</u>

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

<u>admin:</u>

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
}
```

<u>announcement:</u>

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
}
```

<u>event:</u>

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
}
```

article:

The **article** model holds the information (subject and description) for helpful resources/guides. The following JSON string defines an **article** object:

```
{
    "id": auto-created int,
    "subject": 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 - admin created.
302 - admin already exists.
400 - didn't use the correct HTTP Method or didn't have the correct form for the JSON string.

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 - the admin exists and the password is correct.
400 - didn't use the correct HTTP Method.
404 - the admin does not exist.

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:

200 - all admins were returned.

400 - didn't use the correct HTTP 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 - the announcement was created.

302 - the announcement already exists.

400 - incorrect HTTP Method or incorrect JSON string 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:

200 - announcement returned.

400 - incorrect HTTP Method.

404 - announcement doesn't 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:

200 - all announcements returned from database.

400 - incorrect HTTP Method used.

edit-announcement-by-id:

This view updates a specific field for a specific **announcement**.

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

HTTP Method: PATCH

Returns: HTTP Status

Parameters:

id - int

field - str (must be "author", "authored_date", "subject", or "description")

AND a JSON string in the form of

```
{
    "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"
}
```

Troubleshooting:

200 - announcement successfully updated.

400 - incorrect HTTP Method or invalid value for the field parameter.

404 - announcement does not exist with given id.

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 deleted.

400 - incorrect HTTP Method used.

404 - announcement with given id does not exist.