

FEMS Read Only API User Guide

Overview

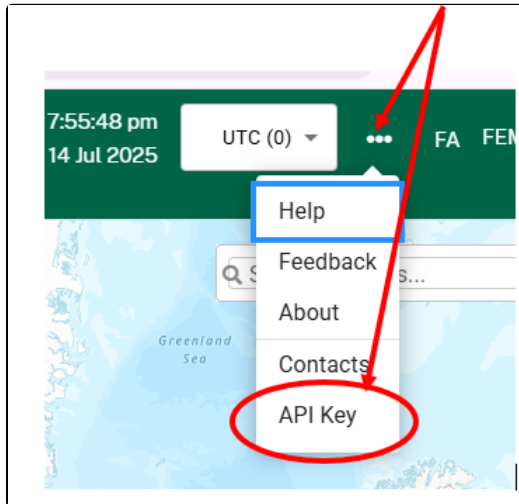
The FEMS Read Only API allows external applications to utilize FEMS data. This guide identifies the instructions for using the FEMS API.

NOTE: Only users with a FEMS API or FEMS Admin role can use the FEMS Read Only API.

FEMS API Key

Follow the instructions in this section to generate and copy a FEMS API Key for use in accessing the read-only FEMS API:

1. Log into FEMS with a user account that has either the FEMS API or FEMS Admin role.
2. Select the hamburger menu  or the ... menu, if the hamburger menu does not display.
3. Select the **API Key** option.



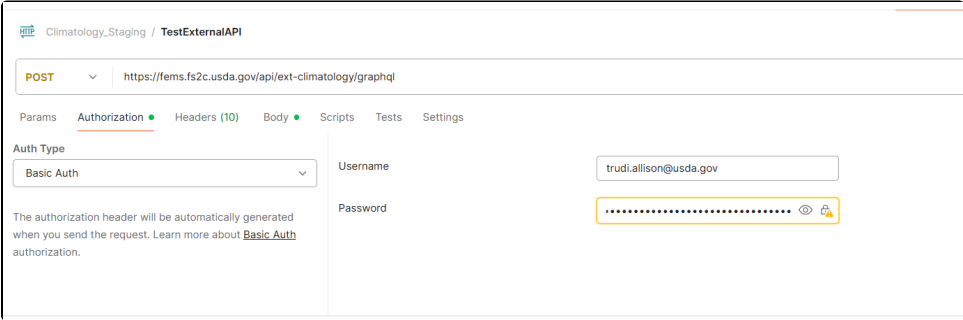
4. When the FEMS API Key Generate window opens, the user name for the account into which you are logged displays.

8. Save the key in a safe location.

NOTE: Each time you click the Generate API Key button, a new API key will be generated for the logged in user account. When a new FEMS API code is generated, the previous code is no longer valid.

Using the FEMS API Key

1. Open an API platform like Postman.
2. Select the Basic Authorization option.
3. Enter the User Name associated with the API key.
4. Enter the API key as the password.



5. Run the API calls for the following url: <https://fems.fs2c.usda.gov/api/ext-climatology/graphql>

Example API Calls For Use with FEMS

API Call	Query	Variables	NOTES
NFDR Catalog	<pre>query NfdrCatalog(\$stationId: String) { nfdrCatalog(returnAll: false stationIds: \$stationId) { _metadata { page per_page total_count page_count } data { fire_danger_model_station_id fire_danger_model_id fire_danger_model_name primary_flg fems_station_id station_id wims_id fuel_model version slope_class } } }</pre>	<pre>{ "stationId": "353213, 55522410" }</pre>	

	<pre>grass_type scm mxd kmdi_parameter_catalog_id dead_fuel_parameter_catalog_id gsi_parameter_catalog_id kmdi_threshold kmdi_initial kmdi_parameter_catalog_name catalog_name herb_days_avg herb_max_gsi herb_greenup herb_max herb_min herb_temp_min_max herb_temp_min_min herb_vpd_min herb_vpd_max herb_day_length_min herb_day_length_max herb_vpd_avg_flg woody_days_avg woody_max_gsi woody_greenup woody_max woody_min woody_temp_min_max woody_temp_min_min woody_vpd_min woody_vpd_max woody_day_length_min woody_day_length_max woody_vpd_avg_flg herb_precip_total_min herb_precip_total_max herb_total_precip_days herb_use_precip_flg woody_precip_total_min woody_precip_total_max woody_total_precip_days woody_use_precip_flg herb_calibration woody_calibration dead_fuel_parameter_catalog_name absorption_rate stick_radius_1_hr_fuel_moisture stick_radius_10_hr_fuel_moisture stick_radius_100_hr_fuel_moisture stick_radius_1000_hr_fuel_moisture modified_time modified_by created_time created_by } } }</pre>		
NFDR Min Max	<pre>query GetNfdrMinMaxWithTime(\$stationIds: String, \$startDate:Date!, \$endDate: Date, \$sortBy: NfdrMinMaxSortBy, \$sortOrder: SortOrder, \$page: Int, \$per_page: Int, \$fuelModel: String, \$zoomLevel: Int, \$hasHistoricData: TriState) { # get nfdr min max nfdrMinMax(stationIds: \$stationIds, startDate: \$startDate,</pre>	<pre>{ "startDate": "2025-07-03", "endDate": "2025-07-24", "stationIds":</pre>	

	<pre> fuelModels: \$fuelModel, zoomLevel: \$zoomLevel, endDate: \$endDate, sortBy: \$sortBy, sortOrder: \$sortOrder, hasHistoricData: \$hasHistoricData, page: \$page, per_page: \$per_page){ _metadata{ page per_page total_count page_count } data{ station_id station_name summary_date nfrd_type fuel_model one_hr_tl_fuel_moisture_min one_hr_tl_fuel_moisture_min_time ten_hr_tl_fuel_moisture_min ten_hr_tl_fuel_moisture_min_time hun_hr_tl_fuel_moisture_min hun_hr_tl_fuel_moisture_min_time thou_hr_tl_fuel_moisture_min thou_hr_tl_fuel_moisture_min_time ignition_component_max ignition_component_max_time spread_component_max spread_component_max_time energy_release_component_max energy_release_component_max_time burning_index_max burning_index_max_time herbaceous_lfi_fuel_moisture woody_lfi_fuel_moisture gsi kbdi quality_code } } } </pre>	<pre> "353213, 55522410", "sortBy": "summary_date", "sortOrder": "DESC", "page": 0, "per_page": 25, "fuelModel": "Y", "hasHistoricData" : "ALL" } </pre>	
NFDRS Obs	<pre> query GetNfrdHourly(\$stationIds: String!, \$fuelModel: String!, \$startDateRange:Date, \$endDateRange: Date, \$startHour:Int, \$endHour:Int, \$dateTimeFormat: DateTimeFormat, \$zoomLevel: Int, \$nfrdType: String, \$hasHistoricData: TriState, \$sortBy: NfrdObsSortBy, \$sortOrder: SortOrder, \$page: Int, \$per_page: Int){ # get nfrd hourly nfrdrsObs(zoomLevel: \$zoomLevel, </pre>	<pre> { "zoomLevel": 5, "startDateRange": "2025-07-14", "endDateRange": " 2025-07-20", "startHour": 0, "endHour": 23, "stationIds": "1708 50, 216901", "page": 0, "per_page": 30000 00, "fuelModel": "Y", "nfrdType": "appen ded", "hasHistoricData": "ALL" } </pre>	

	<pre> stationIds: \$stationIds, fuelModels: \$fuelModel, startDateRange: \$startDateRange, endDateRange: \$endDateRange, startHour: \$startHour, endHour: \$endHour, dateTimeFormat: \$dateTimeFormat, sortBy: \$sortBy, sortOrder: \$sortOrder, page: \$page, per_page: \$per_page, nldrType: \$nldrType, hasHistoricData: \$hasHistoricData,){ _metadata{ page per_page total_count page_count } data{ station_id station_name observation_time observation_time_1st display_hour display_hour_1st nldr_type fuel_model one_hr_tl_fuel_moisture ten_hr_tl_fuel_moisture hun_hr_tl_fuel_moisture thou_hr_tl_fuel_moisture ignition_component spread_component energy_release_component burning_index herbaceous_lfi_fuel_moisture woody_lfi_fuel_moisture gsi kbdi quality_code } } } </pre>		
Get Percentile Avg Min Max	<pre> query GetPercentileAvgMinMax(\$stationId: String! \$fuelModel: FuelModelTypes! \$startYear: ClimatologyYear \$endYear: ClimatologyYear \$startMonthDay: ClimatologyMonthDay \$endMonthDay: ClimatologyMonthDay \$startHour: TimeHour \$endHour: TimeHour \$dateTimeFormat: DateTimeFormat){ percentileAvgMinMax(stationIds: \$stationId fuelModel: \$fuelModel climatology: { startYear: \$startYear endYear: \$endYear startMonthDay: \$startMonthDay </pre>	<pre> { "stationId": "4522 0, 55522410", "fuelModel": "Y", "startYear": "200 5", "endYear": "2022" , "startMonthDay": "01-01", "endMonthDay": " 12-31", "startHour": "0", "endHour": "24", "dateTimeFormat" : "LocalStationTime" } </pre>	

```
        endMonthDay: $endMonthDay
        startHour: $startHour
        endHour: $endHour
        dateTimeFormat: $dateTimeFormat
    }
    page: 0
    per_page: 300000
){
  _metadata {
    page
    per_page
    total_count
    page_count
  }
  data {
    station_id
    day
    energy_release_component_max {
      min
      max
      avg
      stddev
    }
    spread_component_max {
      min
      max
      avg
      stddev
    }
    ignition_component_max {
      min
      max
      avg
      stddev
    }
    burning_index_max {
      min
      max
      avg
      stddev
    }
    one_hr_tl_fuel_moisture_min {
      min
      max
      avg
      stddev
    }
    ten_hr_tl_fuel_moisture_min{
      min
      max
      avg
      stddev
    }
    hun_hr_tl_fuel_moisture_min {
      min
      max
      avg
      stddev
    }
    thou_hr_tl_fuel_moisture_min {
      min
      max
      avg
      stddev
    }
  }
}
```

	<pre> herbaceous_lfi_fuel_moisture_max{ min max avg stddev } woody_lfi_fuel_moisture_max { min max avg stddev } kdbi_max { min max avg stddev } gsi_max { min max avg stddev } } } } </pre>		
Percentile Levels	<pre> query GetPercentileLevels(\$stationId: String! \$fuelModel: FuelModelTypes! \$startYear: ClimatologyYear \$endYear: ClimatologyYear \$startMonthDay: ClimatologyMonthDay \$endMonthDay: ClimatologyMonthDay \$startHour: TimeHour \$endHour: TimeHour \$percentileLevels: String \$dateTimeFormat: DateTimeFormat){ percentileLevels(stationIds: \$stationId fuelModel: \$fuelModel climatology: { startYear: \$startYear endYear: \$endYear startMonthDay: \$startMonthDay endMonthDay: \$endMonthDay startHour: \$startHour endHour: \$endHour dateTimeFormat: \$dateTimeFormat } percentileLevels: \$percentileLevels page: 0 per_page: 300000){ _metadata { page per_page total_count </pre>	<pre> { "stationId": "4522 0, 55522410", "fuelModel": "Y", "startYear": "200 5", "endYear": "2022" , "startMonthDay": "01-01", "endMonthDay": " 12-31", "startHour": "0", "endHour": "24", "dateTimeFormat" : "LocalStationTime", "percentileLevels" : "90,97,3,10" } </pre>	

	<pre> page_count } data { station_id kbdi_max one_hr_tl_fuel_moisture_min ten_hr_tl_fuel_moisture_min hun_hr_tl_fuel_moisture_min thou_hr_tl_fuel_moisture_min ignition_component_max spread_component_max energy_release_component_max woody_lfi_fuel_moisture_max herbaceous_lfi_fuel_moisture_max burning_index_max } } }</pre>		
Station Metadata	<pre> query GetStationMetaData(\$stationId: String!, \$hasHistoricData: TriState) { # Get Station Data stationMetaData(returnAll:false, stationIds: \$stationId, hasHistoricData: \$hasHistoricData,){ _metadata{ page per_page total_count page_count } data{ station_id fems_station_id network_name network_id wrcc_id state period_record_start period_record_stop time_zone time_zone_offset station_name latitude longitude aspect_direction elevation avg_annual_precip agency fuel_models{ slopeclass grasstype } } } }</pre>	<pre> { "stationId": "353213, 55522410", "hasHistoricData" : "ALL" }</pre>	
Weather Obs	<pre> query GetWeatherObs(\$stationId: String! \$startDate: DateTime!, \$endDate:DateTime!, \$zoomLevel:Int, \$hasHistoricData: TriState) { weatherObs(stationIds: \$stationId</pre>	<pre> { "stationId": "336001, 55522410",</pre>	1. To include local station time for this query: Run the Station Metadata query and request the Time Zone and Time Zone offset:

	<pre> startDateTimeRange: \$startDate endDateTimeRange: \$endDate zoomLevel: \$zoomLevel per_page: 100000 hasHistoricData: \$hasHistoricData){ _metadata { page per_page total_count page_count } data { station_id observation_time observation_time_1st display_hour display_hour_1st temperature hourly_precip relative_humidity wind_speed wind_direction peak_gust_speed peak_gust_dir sol_rad snow_flag observation_type } } </pre>	<pre> "zoomLevel": 5, "startDate": "2025-07-14T17:30:00Z", "endDate": "2025-07-20T18:29:59Z", "hasHistoricData" : "ALL" } </pre>	<pre> time_zone time_zone_offset </pre> 2. Add the offset to the weatherObs variables: {"stationId": "204201", "startDateTimeRange": "2025-07-29T09:33:22-06:00", "endDateTimeRange": "2025-07-29T11:03:22-06:00" } 3. Local station time displays in the response as display_hour_1st. UTC time displays in the response as display_hour: "observation_time": "2025-07-14T17:50:24.000Z", "observation_time_1st": "2025-07-14T12:50:24-05:00",
Weighted Winds	<pre> query WeightedWinds { weightedWinds(stationIds: "480403" climatology: { startYear: "2014" endYear: "2023" startMonthDay: "01-01" endMonthDay: "12-31" startHour: "10" endHour: "21" } windType: both){ station_id calmThreshold calmWeight validRecords totalRows windType table } } </pre>		
wxMinMax	<pre> query WxMinMax(\$stationIds: String!, \$startDate:Date!, \$endDate:Date, \$zoomLevel: Int!, \$hasHistoricData: TriState) { wxMinMax(startDate: \$startDate endDate: \$endDate zoomLevel: \$zoomLevel stationIds: \$stationIds hasHistoricData: \$hasHistoricData </pre>	<pre> { "zoomLevel": 5, "startDate": "2025-07-14", "endDate": "2025-07-20", "stationIds": "211401, </pre>	

```
    page: 0
    per_page: 100000
  ){
    _metadata {
      page
      per_page
      total_count
      page_count
    }
    data {
      station_id
      station_name
      summary_date
      observation_type
      temperature_min
      temperature_max
      relative_humidity_min
      relative_humidity_max
      wind_speed_min
      wind_speed_max
      peak_gust_speed_min
      peak_gust_speed_max
      peak_wind_gust_direction
      peak_wind_gust_time
      solar_radiation_max
      daily_precipitation_total
      snow_flag
    }
  }
}
```

```
55522410",
  "hasHistoricData"
: "ALL"
}
```