FEMS Read Only API User Guide

Overview

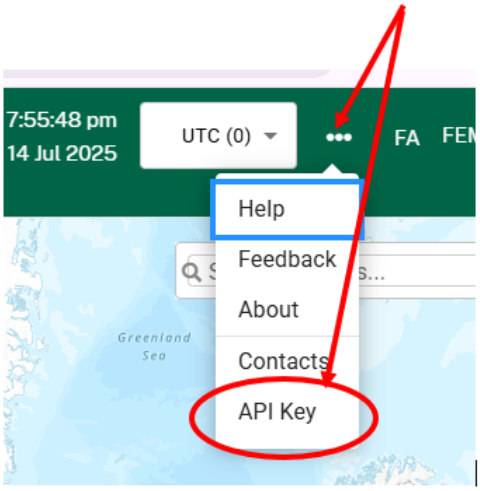
The FEMS Read Only API allows external applications to utilize FEMS data. This guide identifies the instructions for using the FEMS API.

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

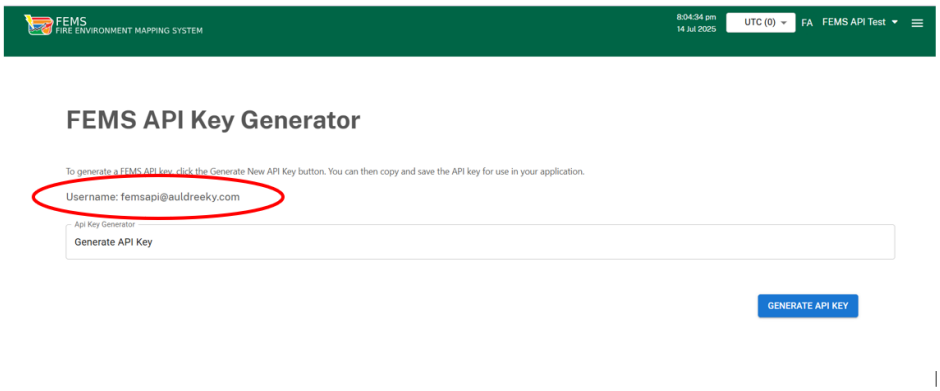
FEMS API Key

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

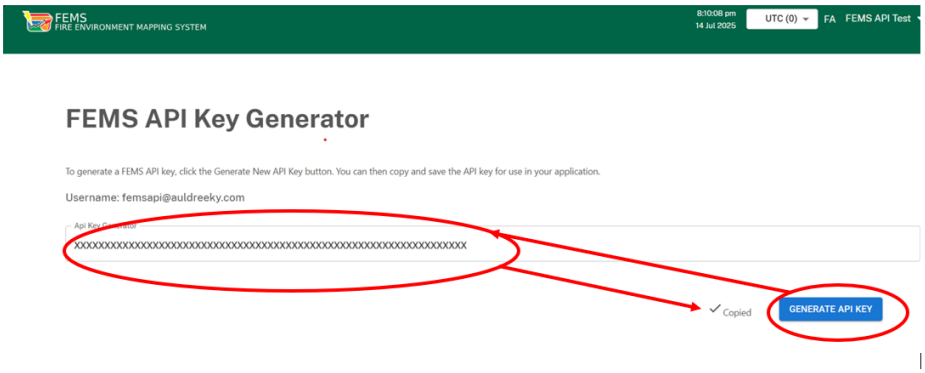
1. Log into FEMS with a user account that has either the FEMS API or FEMS Admin role.
2. Select the hamburger menu or the . . . menu, if the hamburger menu does not display.
3. Select the **API Key** option.



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



5. Click the **Generate API Key** button to generate a new FEMS API key that is linked to the user account in the **Username** field.
6. Click **Continue** when the confirmation message displays.
7. The key is automatically copied to the clipboard on your computer.



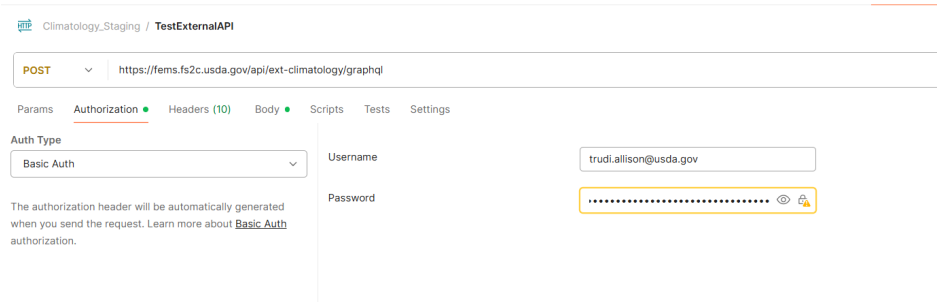
8. Save the key in a safe location.

   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 When a new FEMS API code is generated, the previous code is no longer valid.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	query NfdrCatalog(
	$stationId: String!
	$fuelModels: [FuelModelTypes]
	$returnAll: Boolean
	$fireDangerModelName: String
	$kbdiParameterCatalogName: String
	$catalogName: String
	$page: Int
	$per_page: Int
	) {
	nfdrCatalog(
	returnAll: $returnAll
	stationIds: $stationId
	fuelModels: $fuelModels
	fireDangerModelName: $fireDangerModelName
	kbdiParameterCatalogName: $kbdiParameterCatalogName
	catalogName: $catalogName
	page: $page
	per_page: $per_page
	) {
	_metadata {
	page
	per_page
	total_count
	page_count
	}
	data {
	fire_danger_model_station_id
	fire_danger_model_id
	fire_danger_model_name
		Optional Variables:
		
		returnAll: Boolean
		
		fuelModels: [FuelModelTypes]
		
		fireDangerModelName: String
		
		kbdiParameterCatalogName: String
		
		catalogName: String
		
		page: Int
		
		per_page: Int
		
		Required Variables
		
		stationIds: String
		
		Example GraphQL Variables:
		
		{
		"returnAll": null,
		"stationId": "353213, 55522410",
		"fuelModels": null,
		"fireDangerModelName": null,
		"kbdiParameterCatalogName": null
			Acceptable [FuelModelTypes] values:
			
			null
			
			"V"
			
			"W"
			
			"X"
			
			"Y"
			
			"Z"
			
			Acceptable Boolean values:
			
			null
			
			false
			
			true

	primary_flg fems_station_id station_id wims_id fuel_model version slope_class grass_type scm mxd kbdi_parameter_catalog_id dead_fuel_parameter_catalog_id gsi_parameter_catalog_id kbdi_threshold kbdi_initial kbdi_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	, "catalogName": null, "page": 0, "per_page": 100000 }	

	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 } } }		
NFDR Min Max	query NfdrMinMax($startDate: Date!, $stationIds: String, $wrccIds: String, $fuelModels: String, $nfdrType: String, $zoomLevel: Int, $endDate: Date, $hasHistoricData: TriState, $sortBy: NfdrMinMaxSortBy, $sortOrder: SortOrder, $page: Int, $perPage: Int) { nfdrMinMax(startDate: $startDate, stationIds: $stationIds, wrccIds: $wrccIds, fuelModels: $fuelModels, nfdrType:	**<u>Optional Variables:</u>** stationIds: String wrccIds: String fuelModels: String nfdrType: String	**Acceptable <u>TriState</u> values:** "ALL" "FALSE" "TRUE"

	$nfdrType, zoomLevel: $zoomLevel, endDate: $endDate, hasHistoricData: $hasHistoricData, sortBy: $sortBy, sortOrder: $sortOrder, page: $page, per_page: $perPage) {
	
	_metadata {
	
	page
	
	page_count
	
	per_page
	
	total_count
	
	}
	
	data {
	
	burning_index_max
	
	burning_index_max_time
	
	elevation
	
	energy_release_component_max
	
	energy_release_component_max_time
	
	fuel_model
	
	gsi
	
	herbaceous_lfi_fuel_moisture
	
	hun_hr_tl_fuel_moisture_min
		zoomLevel: Int
		
		endDate: Date
		
		sortBy: NfdrMinMaxSortBy
		
		sortOrder: SortOrder
		
		page: Int
		
		perPage: Int
		
		Required Variables:
		
		startDate: Date!
		
		hasHistoricData: TriState
		
		Example Graphql Query:
		
		{
		"startDate": "2025-07-03",
		"stationIds": null,
		"wrccIds": null,
		"fuelModels": null,
		"nfdrType": null,
		"zoomLevel": null,
		"endDate": "2025-07-04",
		"hasHistoricData": "ALL",
		"sortBy": null,
		"sortOrder": null,
		"page": null,
			Acceptable NfdrMinMaxSortBy Values:
			
			"station_id"
			
			"station_name"
			
			"summary_date"
			
			Acceptable SortOrder Values:
			
			"asc"
			
			"desc"
			
			Date Format:
			
			YYYY-MM-DD

	hun_hr_tl_fuel_moisture_min_time
	
	ignition_component_max
	
	ignition_component_max_time
	
	kbdi
	
	latitude
	
	longitude
	
	nfdr_type
	
	one_hr_tl_fuel_moisture_min
	
	one_hr_tl_fuel_moisture_min_time
	
	quality_code
	
	spread_component_max
	
	spread_component_max_time
	
	station_id
	
	station_name
	
	summary_date
	
	ten_hr_tl_fuel_moisture_min
	
	ten_hr_tl_fuel_moisture_min_time
	
	thou_hr_tl_fuel_moisture_min
	
	thou_hr_tl_fuel_moisture_min_time
		"perPage": null
		}
		
		{
		"startDate": "2026-03-12",
		"endDate": "2026-04-02",
		"stationIds": "102004",
		"sortBy": "summary_date",
		"sortOrder": "DESC",
		"page": 0,
		"per_page": 25,
		"fuelModel": "Y",
		"hasHistoricData": "ALL",
		
		"stationIds": null,
		"wrccIds": null,
		
		"nfdrType": null,
		"zoomLevel": null,
		}	

	woody_lfi_fuel_moisture wrcc_id } } }		
NFDRS Obs	query NfdrsObs($fuelModels: String! $stationIds: String $wrccIds: String $nfdrType: String $zoomLevel: Int $startDateRange: Date $endDateRange: Date $startHour: Int $endHour: Int $dateTimeFormat: DateTimeFormat $hasHistoricData: TriState $sortBy: NfdrObsSortBy $sortOrder: SortOrder $page: Int $perPage: Int) { nfdrsObs(fuelModels: $fuelModels stationIds: $stationIds wrccIds: $wrccIds nfdrType: $nfdrType zoomLevel: $zoomLevel startDateRange: $startDateRange endDateRange: $endDateRange startHour: $startHour endHour: $endHour	**Optional Variables:** stationIds: String wrccIds: String nfdrType: String zoomLevel: Int sortBy: NfdrObsSortBy sortOrder: SortOrder page: Int perPage: Int startHour: Int endHour: Int dateTimeFormat: DateTimeFormat **Required Variables:**	**Acceptable DateTimeFormat values:** "LocalStationTime" "UTC" **Acceptable SortOrder Values:** "asc" "desc" **Acceptable TriState values:** "ALL" "FALSE" "TRUE" **Acceptable NfdrObsSortby values:** "observation_time" "station_id"

	dateTimeFormat: $dateTimeFormat
	hasHistoricData: $hasHistoricData
	sortBy: $sortBy
	sortOrder: $sortOrder
	page: $page
	per_page: $perPage
	) {
	_metadata {
	page
	per_page
	total_count
	page_count
	}
	data {
	station_name
	station_id
	wrcc_id
	latitude
	longitude
	elevation
	observation_time
	observation_time_lst
	display_hour
	display_hour_lst
	masked_observation_time
	nfdr_date
	nfdr_time
	nfdr_type
	fuel_model
	fuel_model_version
	kbdi
	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
		fuelModels: String!
		
		startDateRange: Date
		
		endDateRange: Date
		
		hasHistoricData: TriState
		
		Example GraphQL Query:
		
		{
		"startDateRange": "2025-07-03",
		"endDateRange": "2025-07-04",
		"hasHistoricData": "ALL",
		"sortBy": "",
		"fuelModels": "Y",
		"wrccIds": null,
		"stationIds": null,
		"nfdrType": null,
		"zoomLevel": null,
		"startHour": null,
		"endHour": null,
		"dateTimeFormat": "LocalStationTime",
		"sortOrder": null,
		"page": null,
		"perPage": null,
		}
			Acceptable Date Format:
			
			"YYYY-MM-DD"
			
			Acceptable TimeHour Format
			
			"HH"

	herbaceous_lfi_fuel_moisture woody_lfi_fuel_moisture gsi quality_code } } }		

Get Percentile Avg Min Max	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
	endMonthDay: $endMonthDay
	startHour: $startHour
	endHour: $endHour
	dateTimeFormat: $dateTimeFormat
	}
	page: 0
	per_page: 300000
	) {
	_metadata {
	page
	per_page
	total_count
		Optional Variables:
		
		startYear: ClimatologyYear
		
		endYear: ClimatologyYear
		
		startMonthDay: ClimatologyMonthDay
		
		endMonthDay: ClimatologyMonthDay
		
		endHour: TimeHour
		
		startHour: TimeHour
		
		dateTimeFormat: DateTimeFormat
		
		Required Variables:
		
		fuelModel: FuelModelTypes!
		
		stationId: String!
			Acceptable FuelModelTypes! values:
			
			null
			
			"V"
			
			"W"
			
			"X"
			
			"Y"
			
			"Z"
			
			Acceptable ClimatologyYear format:
			
			"YYYY"
			
			Acceptable ClimatologyMonthDay format:
			
			"MM-DD"
			
			Acceptable TimeHour Format
			
			"HH"

	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
		Example GraphQL Query:
		
		{
		"stationId": "45220, 55522410",
		"fuelModel": "Y",
		"startYear": "2005",
		"endYear": "2022",
		"startMonthDay": "01-01",
		"endMonthDay": "12-31",
		"startHour": "0",
		"endHour": "24",
		"dateTimeFormat": "LocalStation Time"
		}	

	stddev
	}
	hun_hr_tl_fuel_moisture_min {
	min
	max
	avg
	stddev
	}
	thou_hr_tl_fuel_moisture_min {
	min
	max
	avg
	stddev
	}
	herbaceous_lfi_fuel_moisture_max {
	min
	max
	avg
	stddev
	}
	woody_lfi_fuel_moisture_max {
	min
	max
	avg
	stddev
	}
	kbdi_max {
	min
	max
	avg
	stddev
	}
	gsi_max {
	min
	max
	avg
	stddev
	}
	burning_index {		

	min
	max
	avg
	stddev
	}
	energy_release_component {
	min
	max
	avg
	stddev
	}
	gsi {
	min
	max
	avg
	stddev
	}
	herbaceous_lfi_fuel_moisture {
	min
	max
	avg
	stddev
	}
	hun_hr_tl_fuel_moisture {
	min
	max
	avg
	stddev
	}
	ignition_component {
	min
	max
	avg
	stddev
	}
	kbdi {
	min
	max
	avg		

	stddev
	}
	one_hr_tl_fuel_moisture {
	min
	max
	avg
	stddev
	}
	spread_component {
	min
	max
	avg
	stddev
	}
	ten_hr_tl_fuel_moisture {
	min
	max
	avg
	stddev
	}
	thou_hr_tl_fuel_moisture {
	min
	max
	avg
	stddev
	}
	woody_lfi_fuel_moisture {
	min
	max
	avg
	stddev
	}
	}
	}
	}		

Percentile Levels	query GetPercentileLevels($stationId: String! $fuelModel: FuelModelTypes! $startYear: ClimatologyYear	**Optional Variables:**	**Acceptable FuelModelTypes! values:**

	$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
	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
		startYear: ClimatologyYear
		
		endYear: ClimatologyYear
		
		startMonthDay: ClimatologyMonthDay
		
		endMonthDay: ClimatologyMonthDay
		
		endHour: TimeHour
		
		startHour: TimeHour
		
		dateTimeFormat: DateTimeFormat
		
		Required Variables:
		
		fuelModel: FuelModelTypes!
		
		stationId: String!
		
		percentileLevels: String
		
		Example Graphql Variable:
		
		{
		"stationId": "45220, 55522410",
		"fuelModel": "Y",
		"startYear": "2005",
		"endYear": "2022",
		"startMonthDay": "01-01",
		"endMonthDay": "12-31",
		"startHour": "0",
			null
			
			"V"
			
			"W"
			
			"X"
			
			"Y"
			
			"Z"
			
			Acceptable ClimatologyYear format:
			
			"YYYY"
			
			Acceptable ClimatologyMonthDay format:
			
			"MM-DD"
			
			Acceptable TimeHour Format
			
			"HH"

	energy_release_component_max woody_lfi_fuel_moisture_max herbaceous_lfi_fuel_moisture_max burning_index_max } } }	"endHour": "24", "dateTimeFormat": "LocalStation Time", "percentileLevels": "90,97,3,10" }	

Station Metadata	query StationMetaData(
	$returnAll: Boolean
	$zoomLevel: Int
	$stationIds: String
	$networkName: String
	$stateId: String
	$agency: String
	$stationType: String
	$returnList: Boolean
	$sortBy: StationMetadataSortBy
	$sortOrder: SortOrder
	$hasHistoricData: TriState
	$page: Int
	$perPage: Int
	) {
	stationMetaData(
	returnAll: $returnAll
	zoomLevel: $zoomLevel
	stationIds: $stationIds
	networkName: $networkName
	stateId: $stateId
	agency: $agency
	stationType: $stationType
	returnList: $returnList
	hasHistoricData: $hasHistoricData
	sortBy: $sortBy
	sortOrder: $sortOrder
	page: $page
	per_page: $perPage
	) {
	_metadata {
	page
		Optional Variables:
		
		returnAll: Boolean
		
		zoomLevel: Int
		
		stationIds: String
		
		networkName: String
		
		stateId: String
		
		agency: String
		
		stationType: String
		
		returnList: Boolean
		
		sortBy: StationMetadataSortBy
		
		sortOrder: SortOrder
		
		page: Int
		
		perPage: Int
		
		Required Variables
			Acceptable TriState values:
			
			"ALL"
			
			"FALSE"
			
			"TRUE"
			
			Acceptable StationMetadataSortBy Values:
			
			"station_id"
			
			"station_name"
			
			"network_name"
			
			"agency"
			
			"state"
			
			Acceptable SortOrder Values:
			
			"asc"
			
			"desc"

	per_page
	total_count
	page_count
	}
	data {
	fems_station_id
	station_id
	wims_id
	nesdis_id
	wrcc_id
	station_name
	state
	county_name
	latitude
	longitude
	elevation
	slope
	aspect
	aspect_direction
	aspect_degree
	kbdi_threshold
	init_kbdi
	avg_annual_precip
	annual_maintenance_date_time
	period_record_start
	period_record_stop
	station_status
	class
	network_id
	network_name
	agency
	region
	unit
	sub_unit
	ownership_type
	goes
	tx_frequency
	obs_frequency
	site_description
		hasHistoricData: TriState
		
		<u>Example GraphQL Query:</u>
		
		{
		"returnAll": true,
		"zoomLevel": null,
		"stationIds":"353213, 55522410",
		"networkName": null,
		"stateId": null,
		"agency": null,
		"stationType": null,
		"returnList": null,
		"hasHistoricData": "ALL",
		"sortBy": null,
		"sortOrder": null,
		"page": null,
		"perPage": null
		}
		
		{
		"returnAll": true,
		"zoomLevel": null,
		"stationIds": null,
		"networkName": null,
		"stateId": null,
		"agency": null,
		"stationType": null,
		"returnList": null,
		"hasHistoricData": "ALL",
		"sortBy": null,
		"sortOrder": null,
		"page": null,	

	maintenance_standard
	reg_scheduled_observation_time
	time_zone
	time_zone_offset
	zoom_level
	transmit_time
	modified_time
	modified_by
	created_time
	created_by
	fuel_models {
	fuel_model
	version
	primary_flg
	slopeclass
	grasstype
	scm
	extmoi
	last_modified_date
	}
	first_observation
	last_observation
	site
	observing_agency
	gsi_parameter_catalog_id
	station_type
	nfdr_visibility
	station_metadata_visible
	has_historic_data
	weather_daily_visible
	nfdrs_daily_visible
	weather_hourly_visible
	nfdrs_hourly_visible
	weather_daily_download
	weather_hourly_download
	nfdrs_daily_download
	nfdrs_hourly_download
	produce_nfdrs
	ingest_observation
		"perPage": null
		}	

	} } }		

Weather Obs	query WeatherObs(
	$startDateTimeRange: DateTime!
	$endDateTimeRange: DateTime!
	$stationIds: String
	$wrccIds: String
	$attribute: String
	$timeZone: String
	$zoomLevel: Int
	$hasHistoricData: TriState
	$sortBy: WxObsSortBy
	$sortOrder: SortOrder
	$page: Int
	$perPage: Int
	) {
	weatherObs(
	startDateTimeRange: $startDateTimeRange
	endDateTimeRange: $endDateTimeRange
	stationIds: $stationIds
	wrccIds: $wrccIds
	attribute: $attribute
	timeZone: $timeZone
	zoomLevel: $zoomLevel
	hasHistoricData: $hasHistoricData
	sortBy: $sortBy
	sortOrder: $sortOrder
	page: $page
	per_page: $perPage
	) {
	_metadata {
	page
	per_page
	total_count
	page_count
	}
		<u>Optional Variables:</u>
		
		wrccIds: String
		
		zoomLevel: Int
		
		stationIds: String
		
		attribute: String
		
		timeZone: String
		
		hasHistoricData: TriState
		
		sortBy: WxObsSortBy
		
		sortOrder: SortOrder
		
		page: Int
		
		perPage: Int
		
		<u>Required Variables</u>
		
		startDateTimeRange: DateTime!
		
		endDateTimeRange: DateTime!
		
		<u>Example Graphql Query:</u>
			1. To include local station time for this query:
			Run the Station Metadata query and request the Time Zone and Time Zone offset:
			
			time_zone
			time_zone_offset
			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_lst. UTC time displays in the response as display_hour:
			
			"observation_time": "2025-07-14T17:50:24.000Z",
			"observation_time_lst": "2025-07-14T12:50:24-05:00",
			
			Acceptable <u>WxObsSortBy</u> Values:
			
			"station_id"
			
			"station_name"
			
			"observation_time"
			
			"temperature_min"

	data {
	station_id
	wrcc_id
	station_name
	latitude
	longitude
	elevation
	zoom_level
	station_type
	observation_time
	observation_time_lst
	display_hour
	display_hour_lst
	masked_observation_time
	display_date
	temperature
	relative_humidity
	hourly_precip
	wind_speed
	wind_direction
	peak_gust_speed
	peak_gust_dir
	sol_rad
	snow_flag
	observation_type
	hex
	t_flag
	rh_flag
	pcp_flag
	ws_flag
	wa_flag
	sr_flag
	gs_flag
	ga_flag
	}
	}
	}
		{
		"startDateTimeRange": "2026-03-26T16:30:00Z",
		"endDateTimeRange": "2026-03-26T17:29:59Z",
		"perPage": null,
		"page": null,
		"sortOrder": null,
		"sortBy": null,
		"hasHistoricData": null,
		"zoomLevel": null,
		"timeZone": null,
		"attribute": null,
		"wrccIds": null,
		"stationIds": null
		}
		
		{
		"stationId": "336001, 55522410",
		"zoomLevel": 5,
		"startDate": "2025-07-14T17:30:00Z",
		"endDate": "2025-07-20T18:29:59Z",
		"hasHistoricData": "ALL"
		}
			"temperature"
			
			"relative_humidity"
			
			"hourly_precip"
			
			"wind_speed"
			
			"wind_direction"
			
			"peak_gust_speed"
			
			"peak_gust_direction"
			
			"sol_rad"
			
			Acceptable SortOrder Values:
			
			"asc"
			
			"desc"

Weighted Winds	query WeightedWinds(
	$stationIds: String!
	$startYear: ClimatologyYear
	$endYear: ClimatologyYear
	$startMonthDay: ClimatologyMonthDay
	$endMonthDay: ClimatologyMonthDay
	$startHour: TimeHour
	$endHour: TimeHour
	$dateTimeFormat: DateTimeFormat
	$windType: WindType
	) {
	weightedWinds(
	stationIds: $stationIds
	climatology: {
	startYear: $startYear
	endYear: $endYear
	startMonthDay: $startMonthDay
	endMonthDay: $endMonthDay
	startHour: $startHour
	endHour: $endHour
	dateTimeFormat: $dateTimeFormat
	}
	windType: $windType
	) {
	station_id
	calmThreshold
	calmWeight
	validRecords
	totalRows
	windType
	table
	}
	}
		Optional Variables:
		
		startYear: ClimatologyYear
		
		endYear: ClimatologyYear
		
		startMonthDay: ClimatologyMonthDay
		
		endMonthDay: ClimatologyMonthDay
		
		endHour: TimeHour
		
		startHour: TimeHour
		
		dateTimeFormat: DateTimeFormat
		
		windType: WindType
		
		Required Variables
		
		stationIds: String!
		
		Example Graphql Query:
		
		{
		"stationIds": "102004",
		"windType": "avg",
		"startYear": "2005",
		"endYear": "2022",
		"startMonthDay": "01-01",
		"endMonthDay": "12-31",
		"startHour": "0",
			Acceptable ClimatologyYear format:
			
			"YYYY"
			
			Acceptable ClimatologyMonthDay format:
			
			"MM-DD"
			
			Acceptable TimeHour Format
			
			"HH"
			
			Acceptable windType Values:
			
			"both"
			
			"avg"
			
			"gust"
			
			Acceptable DateTimeFormat values:
			
			"LocalStationTime"
			
			"UTC"

		"endHour": "24", "dateTimeFormat": "LocalStation Time" }	

wxMinMax	query WxMinMax(
	$startDate: Date!
	$stationIds: String
	$wrccIds: String
	$endDate: Date
	$zoomLevel: Int
	$hasHistoricData: TriState
	$sortBy: WxMinMaxSortBy
	$sortOrder: SortOrder
	$page: Int
	$perPage: Int
	) {
	wxMinMax(
	startDate: $startDate
	stationIds: $stationIds
	wrccIds: $wrccIds
	endDate: $endDate
	zoomLevel: $zoomLevel
	hasHistoricData: $hasHistoricData
	sortBy: $sortBy
	sortOrder: $sortOrder
	page: $page
	per_page: $perPage
	) {
	_metadata {
	page
	per_page
	total_count
	page_count
	}
	data {
	station_name
	station_id
		Optional Variables:
		
		wrccIds: String
		
		zoomLevel: Int
		
		stationIds: String
		
		sortBy: WxMinMaxSortBy
		
		sortOrder: SortOrder
		
		page: Int
		
		perPage: Int
		
		endDate: Date
		
		Required Variables
		
		startDate: Date!
		
		hasHistoricData: TriState
		
		Example Graphql Query:
		
		{
		"startDate": "2026-03-25",
			Acceptable WxMinMaxSortBy Values:
			
			"station_id"
			
			"station_name"
			
			"summary_date"
			
			"temperature_min"
			
			"temperature_max"
			
			"relative_humidity_min"
			
			"relative_humidity_max"
			
			"daily_precipitation_total"
			
			"wind_speed_min"
			
			"wind_speed_max"
			
			"peak_gust_speed_min"
			
			"peak_gust_speed_max"
			
			"peak_wind_gust_direction"
			
			"solar_radiation_max"

	wrcc_id latitude longitude elevation 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 } } }	"stationIds": null, "wrccIds": null, "endDate": null, "zoomLevel": null, "hasHistoricData": "ALL", "sortBy": null, "sortOrder": null, "page": null, "perPage": null }	**Acceptable SortOrder Values:** "asc" "desc" **Acceptable TriState values:** "ALL" "FALSE" "TRUE" **Date Format:** YYYY-MM-DD