

# Clinical Data API Documentation

---

## Overview

The Clinical Data API allows clients to retrieve detailed clinical data associated with a specific gait process identified by a `process_id`. This includes gait parameters, phase-specific joint angles, statuses, and URLs for corresponding frames. This API is designed for use in gait analysis applications to provide insights into various gait phases and the overall gait health status.

---

## Endpoint

**URL:** `/clinicalApi/?process_id={process_id}`

**Example:** `/clinical_data/?process_id=55-webportal-55-gait-1731312374351924`

**Method:** `GET`

**Parameters:**

- `process_id` (string, required): A unique identifier for the gait process.
- 

## API Behavior

The API accepts a `GET` request with the `process_id` as a query parameter. It retrieves the `ClinicalData` object associated with the provided `process_id` and returns the details in JSON format.

If the `process_id` parameter is missing or invalid, the API responds with an error.

---

## Response

### Success Response

If the `process_id` is valid and the corresponding `ClinicalData` object exists, the API responds with a `200 OK` status and returns the following JSON structure:

```
{
  "test_id": "string",
  "test_date": "datetime",
  "owner": "string",

  // Gait Parameters
  "step_length": "float",
  "stride_length": "float",
  "speed": "float",
  "cadence": "float",

  // Gait Parameters Status
```

```
"step_length_status": "string",
"stride_length_status": "string",
"speed_status": "string",
"cadence_status": "string",

// Heel Strike Phase
"hs_frame": "integer",
"hs_hip": "float",
"hs_knee": "float",
"hs_ankle": "float",
"hs_status": "string",
"hs_frame_url": "string",

// Midstance Phase
"ms_frame": "integer",
"ms_hip": "float",
"ms_knee": "float",
"ms_ankle": "float",
"ms_status": "string",
"ms_frame_url": "string",

// Toe Off Phase
"toe_frame": "integer",
"toe_hip": "float",
"toe_knee": "float",
"toe_ankle": "float",
"toe_status": "string",
"toe_frame_url": "string",

// Mid Swing Phase
"sw_frame": "integer",
"sw_hip": "float",
"sw_knee": "float",
"sw_ankle": "float",
"sw_status": "string",
"sw_frame_url": "string"
}
```

## Field Descriptions

### Clinical Data Attributes

**test\_id**: Unique identifier for the test (string). **test\_date**: Date and time when the test was conducted. **owner**: Username of the user associated with this clinical data. **Gait Parameters** **step\_length**: Distance between successive steps (float, inches). **stride\_length**: Distance covered in a full gait cycle (float, inches). **speed**: Speed of walking (float, feet per second). **cadence**: Steps taken per minute (float). **Gait Parameters Status** Indicates whether the respective gait parameter is normal or abnormal:

**step\_length\_status**: Status of step length. **stride\_length\_status**: Status of stride length. **speed\_status**: Status of speed. **cadence\_status**: Status of cadence. **Gait Phases**

Each gait phase contains data on frame number, joint angles (hip, knee, ankle), and frame URL.

- Heel Strike Phase:

**hs\_frame**: Frame number where heel strike occurs. **hs\_hip**, **hs\_knee**, **hs\_ankle**: Hip, knee, and ankle angles during heel strike. **hs\_status**: Status indicating normal or abnormal heel strike. **hs\_frame\_url**: URL linking to the image or video frame of heel strike.

- Midstance Phase:

**ms\_frame**: Frame number where midstance occurs. **ms\_hip**, **ms\_knee**, **ms\_ankle**: Hip, knee, and ankle angles during midstance. **ms\_status**: Status of the midstance phase. **ms\_frame\_url**: URL linking to the image or video frame of midstance.

- Toe Off Phase:

**toe\_frame**: Frame number where toe-off occurs. **toe\_hip**, **toe\_knee**, **toe\_ankle**: Hip, knee, and ankle angles during toe-off. **toe\_status**: Status indicating normal or abnormal toe-off. **toe\_frame\_url**: URL linking to the image or video frame of toe-off.

- Mid Swing Phase:

**sw\_frame**: Frame number where mid swing occurs. **sw\_hip**, **sw\_knee**, **sw\_ankle**: Hip, knee, and ankle angles during mid swing. **sw\_status**: Status indicating normal or abnormal mid swing. **sw\_frame\_url**: URL linking to the image or video frame of mid swing.

## Error Response

If the `process_id` is missing or incorrect, the API responds with a 400 Bad Request status and an error message:

```
{  "error": "Missing process_id parameter"}
```

## Example Request

GET /api/clinical\_data/?process\_id=55-webportal-55-gait-1731312374351924

## Code example

---

- API DECLARE

```
class ClinicalData(models.Model):
    test_id = models.CharField(max_length=20, unique=True, default='')
    test_date = models.DateTimeField(auto_now_add=True)

    owner = models.ForeignKey(User,
                              related_name='%(class)s_related',
                              default=None,
                              on_delete=models.CASCADE)

    # Foreign key to SagittalGait
    process = models.ForeignKey(
        ProcessGait,
        on_delete=models.CASCADE,
        related_name='clinical_data',
        primary_key=True
    )

    # Gait Parameters
    step_length = models.DecimalField(
        max_digits=5,
        decimal_places=2,
        default=0,
        help_text="Step length in inches"
    )
    stride_length = models.DecimalField(
        max_digits=5,
        decimal_places=2,
        default=0,
        help_text="Stride length in inches"
    )
    speed = models.DecimalField(
        max_digits=4,
        decimal_places=2,
        default=0,
        help_text="Speed in ft/sec"
    )
    cadence = models.DecimalField(
        max_digits=5,
        decimal_places=2,
        default=0,
        help_text="Cadence in steps/min"
    )

    # Gait Parameters Status
```

```
    step_length_status = models.CharField(max_length=20,
default='',blank=True)
    stride_length_status = models.CharField(max_length=20,
default='',blank=True)
    speed_status = models.CharField(max_length=20, default='',blank=True)
    cadence_status = models.CharField(max_length=20, default='',blank=True)

    # Heel Strike Phase
    hs_frame = models.IntegerField(help_text="Heel Strike frame number",
default=0)
    hs_hip = models.DecimalField(max_digits=6, decimal_places=2, default=0)
    hs_knee = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    hs_ankle = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    hs_status = models.CharField(max_length=20,
default='Normal',blank=True)
    hs_frame_url = models.URLField(max_length=60, default='',blank=True)

    # Midstance Phase
    ms_frame = models.IntegerField(help_text="Midstance frame number",
default=0)
    ms_hip = models.DecimalField(max_digits=6, decimal_places=2, default=0)
    ms_knee = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    ms_ankle = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    ms_status = models.CharField(max_length=20,
default='Normal',blank=True)
    ms_frame_url = models.URLField(max_length=60, default='',blank=True)

    # Toe Off Phase
    toe_frame = models.IntegerField(help_text="Toe Off frame number",
default=0)
    toe_hip = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    toe_knee = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    toe_ankle = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    toe_status = models.CharField(max_length=20,
default='Normal',blank=True)
    toe_frame_url = models.URLField(max_length=60, default='',blank=True)

    # Mid Swing Phase
    sw_frame = models.IntegerField(help_text="Mid Swing frame number",
default=0)
    sw_hip = models.DecimalField(max_digits=6, decimal_places=2, default=0)
    sw_knee = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    sw_ankle = models.DecimalField(max_digits=6, decimal_places=2,
default=0)
    sw_status = models.CharField(max_length=20,
default='Normal',blank=True)
```

```
sw_frame_url = models.URLField(max_length=60, default='', blank=True)

class Meta:
    verbose_name = "Clinical Data"
    verbose_name_plural = "Clinical Data"
    ordering = ['-test_date']

def __str__(self):
    return f"Process ID: {self.process.processId}"

@property
def overall_status(self):
    statuses = [
        self.step_length_status,
        self.stride_length_status,
        self.speed_status,
        self.cadence_status,
        self.hs_status,
        self.ms_status,
        self.to_status,
        self.sw_status
    ]
    return "Abnormal" if "Abnormal" in statuses else "Normal"
```

#### - Api Call

```
def api(request):
    print("test api")

    # Retrieve `process_id` from the request's query parameters
    process_id = request.GET.get('process_id')
    process_id = "55-webportal-55-gait-1731312374351924"
    if not process_id:
        return HttpResponse("Missing process_id parameter", status=400)

    # Retrieve ClinicalData object based on `process_id`
    clinical_data = get_object_or_404(ClinicalData, process=process_id)

    # print("Clinical Data Retrieved:", clinical_data)
    # return HttpResponse(f"API is working. Clinical Data:
    {clinical_data.cadence}")

    # Serialize the ClinicalData object to JSON, including all fields
    clinical_data_json = {
        "test_id": clinical_data.test_id,
        "test_date": clinical_data.test_date.isoformat(),
        "owner": clinical_data.owner.username,

        # Gait Parameters
        "step_length": float(clinical_data.step_length),
        "stride_length": float(clinical_data.stride_length),
        "speed": float(clinical_data.speed),
```

```
"cadence": float(clinical_data.cadence),

# Gait Parameters Status
"step_length_status": clinical_data.step_length_status,
"stride_length_status": clinical_data.stride_length_status,
"speed_status": clinical_data.speed_status,
"cadence_status": clinical_data.cadence_status,

# Heel Strike Phase
"hs_frame": clinical_data.hs_frame,
"hs_hip": float(clinical_data.hs_hip),
"hs_knee": float(clinical_data.hs_knee),
"hs_ankle": float(clinical_data.hs_ankle),
"hs_status": clinical_data.hs_status,
"hs_frame_url": clinical_data.hs_frame_url,

# Midstance Phase
"ms_frame": clinical_data.ms_frame,
"ms_hip": float(clinical_data.ms_hip),
"ms_knee": float(clinical_data.ms_knee),
"ms_ankle": float(clinical_data.ms_ankle),
"ms_status": clinical_data.ms_status,
"ms_frame_url": clinical_data.ms_frame_url,

# Toe Off Phase
"toe_frame": clinical_data.toe_frame,
"toe_hip": float(clinical_data.toe_hip),
"toe_knee": float(clinical_data.toe_knee),
"toe_ankle": float(clinical_data.toe_ankle),
"toe_status": clinical_data.toe_status,
"toe_frame_url": clinical_data.toe_frame_url,

# Mid Swing Phase
"sw_frame": clinical_data.sw_frame,
"sw_hip": float(clinical_data.sw_hip),
"sw_knee": float(clinical_data.sw_knee),
"sw_ankle": float(clinical_data.sw_ankle),
"sw_status": clinical_data.sw_status,
"sw_frame_url": clinical_data.sw_frame_url
}

return JsonResponse(clinical_data_json, status=200)
````
```