


אתר גולדפיש – Goldfish


מבוא לשימוש באתר גולדפיש ומבנהו 

סוגי שאילות 


Heat Map •

Marker Map •

Planes •

הרצת שאילות לבדיקה וצפייה בפלט 

שמירת שאילות למאגר ולהרצה ב"Lion Fish" 

ניקוי התיבות באתר 

מבוא לשימוש באתר גולדפיש ומבנהו

אתר "Goldfish" מאפשר כתיבת, בדיקת והוספת שאילתות למאגר השאילתות שניתן לבחור להריץ דרך אתר "Lionfish".

תמונה 1: מבנה אתר "Goldfish"

כפי שניתן לראות בתמונה 1, המציגה את מבנה אתר "Goldfish", באתר נוכל להגדיר את השאילתה (בתיבה הגדולה "Query"), את שם השאילתה (בתיבת "Query Name"), את סוג השאילתה (בתיבת הרשימה הנפתחת "Type") ואת הפרמטרים לשאילתה ואת סוגם (בתיבת "Parameters").

נשים לב! כיצד נסמן פרמטר param מסוג type, שהערך שנרצה לתת לו עבור הרצת השאילתה לבדיקה באתר Goldfish הוא value?
בהגדרת השאילתה, בתיבת Query, נסמן את הפרמטר ע"י הקפתו ב - \$ באופן הבא:
\$.param\$

בהגדרת הפרמטרים לשאילתה, סוגם וערך לדוגמה להרת השאילתה, בתיבת Parameters, נכתוב:
param: type=value

* עבור ריבוי פרמטרים, נכתוב שתיבת Parameters לדוגמה:

name : String = "John", age : Number = 20

* סוג הפרמטר type חייב להיות אחד מארבעת האפשרויות הבאות: Number, String, Date, Time.

תזכורת להערות אלו מופיעה באתר בלחיצה על אייקון סימן השאלה:

דוגמאות מפורטות מופיעות מטה.

סוגי שאילות

Heat Map •

שאילתה מסוג Heat Map מחזירה את עמודות :

○ strength – מספר שמציין את עוצמת החום

○ lat – מספר שמציין קו רוחב

○ lon – מספר שמציין קו אורך

בנקודות הנבחרות לפי השאילתה.

בהרצת השאילתה תוצג על המפה מפת חום צבעונית המשקפת לפי הצבע את עוצמת החום

בנקודות עם קווי הרוחב והאורך שנבחרו.

לדוגמה :

תמונה 2 : דוגמה ליצירת שאילתה מסוג Heat Map

הגדרנו שאילתה שמטרתה להחזיר נתונים גיאוגרפיים ממאגר הנתונים כדי ליצור מפת חום (heat map). היא פונה לטבלה בשם fast_params ובוחרת את העמודות :

○ tele_pp_lat כערך קו רוחב .lat

○ tele_pp_long כערך קו אורך .lon

○ tele_rpm כערך עוצמת החום strength.

השאילתה מססנת את הנתונים לפי המזהה fid. כל הנתונים בטבלה עם המזהה הזה יוחזרו.

כמו כן, מסננים החוצה רשומות שבהן הערכים של קו רוחב (tele_pp_lat) וקו אורך (tele_pp_long) הם אפס.

הנתונים המוחזרים הם רשימה של נקודות גיאוגרפיות (lat, lon) יחד עם ערך strength, שמאפשר להשתמש במידע ליצירת מפת חום שתוצג על המפה באתר Lionfish בעת הרצת השאילתה.

נשים לב שניתן לראות שבאתר הוזן כי שם השאילתה הוא "RPM_FOR_FID", הסוג שלה הוא "Heat Map" והיא משתמשת בפרמטר fid מסוג Number שבהרצתה לדוגמה של השאילתה באתר Goldfish, יקבל את הערך 4.

• Marker Map

שאלתה מסוג Marker Map מחזירה את עמודות :

○ content – הטקסט שיופיע כשנעמוד על הסימון שיופיע על גבי המפה באתר Lionfish

כשנרץ את השאלתה

○ lat – מספר שמציין קו רוחב

○ lon – מספר שמציין קו אורך

בנקודות הנבחרות לפי השאלתה.

בהרצת השאלתה יוצגו על המפה סימוני נקודות ציון בנקודות עם קווי הרוחב והאורך שנבחרו,

שבעמידה עליהם יופיע הטקסט המתאים שנבחר.

לדוגמה :

Goldfish Query Editor

Query Name: START_END_FOR_FID

Type: Marker Map

Parameters: fid : Number =2

Query:

```
marker_map(query: "(select slow_params.tele_pp_lat as lat,slow_params.tele_pp_long as lon,'start: fid=$fid$' as content from slow_params where slow_params.fid = $fid$ and slow_params.tele_pp_lat!=0 and slow_params.tele_pp_long!=0 order by packet asc limit 1) UNION ALL (select slow_params.tele_pp_lat as lat,slow_params.tele_pp_long as lon,'start: fid=$fid$' as content from slow_params where slow_params.fid = $fid$ and slow_params.tele_pp_lat!=0 and slow_params.tele_pp_long!=0 order by packet desc limit 1)") {lat lon content}
```

Output:

```
{}
```

Run

תמונה 3 : דוגמה לשאלתה מסוג Marker Map

הגדרנו שאלתה שמטרתה להציג על המפה סימונים של הנקודות ההתחלתיות והסופיות של

טיסות. השאלתה פונה לטבלת slow_params ובוחרת את העמודות :

○ tele_pp_lat כערך קו רוחב lat.

○ tele_pp_long כערך קו אורך lon.

○ 'start: fid=\$fid\$' as content יוצר מחרוזת תוכן שמציינת את המזהה של הטיסה,

ומשמשת לתיאור הסימון על המפה.

השאלתה מורכבת משתי תת-שאלות המחוברות ב UNION ALL :

○ תת-שאלתה הראשונה בוחרת את הנקודה הראשונה לפי סדר עולה של השדה packet

עם קו רוחב וקו אורך שאינם אפס מתוך הטבלה slow_params עבור המזהה \$fid\$.

○ תת-שאלתה השנייה, בוחרת את הנקודה האחרונה לפי סדר יורד של השדה packet עם

קו רוחב וקו אורך חוקיים מתוך אותה טבלה ועבור אותו מזהה.

השאלתה מסננת את הנתונים בכל אחת מתת-השאלות ע"י הבדיקה שקו הרוחב וקו האורך

אינם אפס. כמו כן, התת-שאלות מוגבלות לשורה אחת כל אחת, (limit 1) כך שהשאלתה

מחזירה רק את הנקודה הראשונה והנקודה האחרונה עבור אותו מזהה.

הנתונים המוחזרים כוללים את קו הרוחב, קו האורך ותיאור התוכן, ('start: fid=\$fid\$') שמציג את הנקודה ההתחלתית והסופית של הטיסה עם סמן על גבי המפה באתר Lionfish (בעת הרצת השאילתה באתר).

נשים לב שניתן לראות שבאתר הוזן כי שם השאילתה הוא "START_END_FOR_FID", הסוג שלה הוא "MARKER_MAP" והיא משתמשת בפרמטר fid מסוג Number שבהרצה לדוגמה של השאילתה באתר Goldfish, יקבל את הערך 2.

Planes

שאלתה מסוג Heat Map מחזירה את עמודות :

- fid – מספר המזהה את הטיסה
- start – מספר המציין זמן התחלה
- end – מספר המציין זמן סיום

עבור הטיסות והזמנים הנבחרים לפי השאלתה.

בהרצת השאלתה יוצגו על המפה איורים של הכטמ"מים (המתאימים לfid'ים) בנקודת זמן ששייכת לפרק הזמן שבין זמן ההתחלה start לזמן הסיום end, לפי נקודת הזמן שנבחרת ע"י slider באתר "Lionfish".

לדוגמה :

Goldfish Query Editor

Query Name: GET_FLIGHT

Type: Planes

Parameters: fid : Number =2

Query:

```
query{
  get_flights(query: "select fid, recording_start as start,
    recording_end as end from metadata where fid=$fid$") {fid start end}
}
```

Output:

```
{}
```

Run

תמונה 4 : דוגמה לשאלתה מסוג Planes

הגדרנו שאלתה שמטרתה לשאוב נתוני טיסה מתחילת ההקלטה עד סופה עבור מזהה ייחודי (fid) של טיסה מסויימת. השאלתה פונה לטבלת metadata ובוחרת את העמודות :

- fid כערך מזהה ייחודי של הטיסה (fid)
- recording_start כערך זמן ההתחלה (start)
- recording_end כערך זמן הסיום (end)

השאלתה מסננת את הנתונים לפי המזהה הייחודי (fid), כך שרק הנתונים של הטיסה עם fid המתאים מוחזרים.

הנתונים המוחזרים כוללים את מזהה הטיסה, זמני התחלת הקלטה וסיומה, כך שיוצג על המפה באתר Lionfish איור של כטמ"מ לאורך הטיסה לפי נקודת הזמן הנבחרת באתר Lionfish בתוך פרק זמן זה.

נשים לב שניתן לראות שבאתר הוזן כי שם השאלתה הוא "GET_FLIGHT", הסוג שלה הוא "PLANE" והיא משתמשת בפרמטר fid מסוג Number שבהרצה לדוגמה של השאלתה באתר Goldfish, יקבל את הערך 2.

הרצת שאילתה לבדיקה וצפייה בקלט

על מנת להריץ שאילתה ולבדוק אותה, ראשית נמלא את התיבות :
"Query Name", "Type", "Parameters" ו- "Query" ולאחר מכן נלחץ על "Run".

אם הרצת השאילתה הצליחה, תופיע הודעת אישור: "Query succeed!" במלבן ירוק בצד ימין למטה. בנוסף, פלט ההרצה יופיע בתיבת "Output".

The screenshot shows the Goldfish Query Editor interface. At the top, there's a yellow header with the logo and "Goldfish Query Editor". Below the header, there are three input fields: "Query Name" (RPM_FOR_FID), "Type" (Heat Map), and "Parameters" (fid : Number =4). To the right of these fields are "Clear" and "Save" buttons. Below the input fields, there are two main panels. The left panel, labeled "Query", contains a SQL query:

```
query{
  heat_map(query: "select tele_pp_lat as lat,tele_pp_long as
lon,tele_rpm as strength from fast_params where fid=$fid$ and
tele_pp_lat!=0 and tele_pp_long!=0") {lat lon strength}
}
```

. The right panel, labeled "Output", displays the JSON response:

```
{
  "heat_map": [
    {
      "__typename": "HeatMapPoint",
      "lat": 0.55661786,
      "lon": 0.6053782,
      "strength": 7436
    },
    {
      "__typename": "HeatMapPoint",
      "lat": 0.55661786,
      "lon": 0.6053782,
      "strength": 7442
    },
    {
      "__typename": "HeatMapPoint",

```

. At the bottom left, there's a "Run" button and a "Status: Done" indicator. At the bottom right, there's a green notification box that says "Query succeed!".

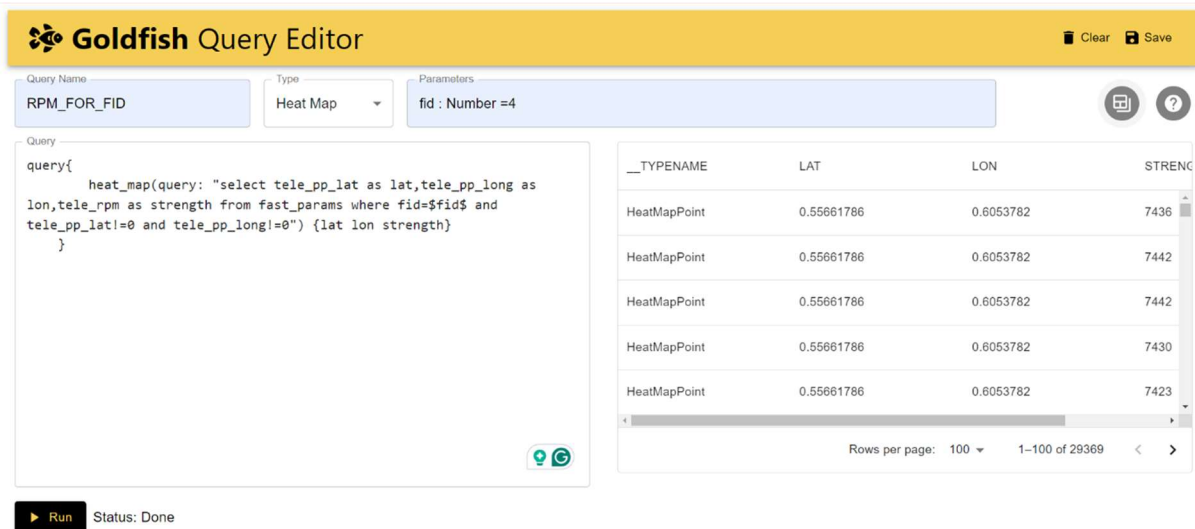
תמונה 5 : שאילתה שהורצה בהצלחה

במקרה של הצלחה, לחיצה על אייקון הסוגריים המסולסלים : יאפשר לנו לצפות בפלט ההרצה גם במבנה של טבלה :

This screenshot is identical to the previous one, showing the Goldfish Query Editor interface with the same query and output. The only difference is that the icon for switching to table view (a pair of curly braces) is highlighted with a red circle, indicating that clicking it will change the output format from JSON to a table.

תמונה 6 : לחיצה על אייקון הסוגריים המסולסלים (לאחר הרצת שאילתה שהצליחה)

לאחר הלחיצה הוא אייקון הסוגריים המסולסלים יתחלף לאייקון הטבלה: . לחיצה על אייקון הטבלה תסגור את טבלת הפלט.



Goldfish Query Editor Clear Save

Query Name: RPM_FOR_FID Type: Heat Map Parameters: fid : Number =4

Query:

```
query{
  heat_map(query: "select tele_pp_lat as lat,tele_pp_long as
lon,tele_rpm as strength from fast_params where fid=$fid$ and
tele_pp_lat!=0 and tele_pp_long!=0") {lat lon strength}
}
```

__TYPENAME	LAT	LON	STRENGTH
HeatMapPoint	0.55661786	0.6053782	7436
HeatMapPoint	0.55661786	0.6053782	7442
HeatMapPoint	0.55661786	0.6053782	7442
HeatMapPoint	0.55661786	0.6053782	7430
HeatMapPoint	0.55661786	0.6053782	7423

Rows per page: 100 1-100 of 29369

Run Status: Done

תמונה 7: פלט שאילתה שהרצתה הצליחה בטבלה ולחיצה על אייקון הטבלה לסגירת הטבלה

אם ההרצה לא הצליחה, תופיע הודעת שגיאה: "Error running query" במלבן אדום בצד ימין למטה. בנוסף, תיארור מפורט של השגיאה יופיע בתיבת "Output".



Goldfish Query Editor Clear Save

Query Name: RPM_FOR_FID Type: Heat Map Parameters: fid : Number =4

Query:

```
query{
  heat_map(query: "select tele_pp_lat as lat,tele_pp_long as
lon,tele_rpm as strength from fast_params where fid=$fid$ and
tele_pp_lat!=0 and tele_pp_long!=0") {lat lon strength}
}{
}
```

Output:

```
{
  "message": "Syntax Error: Expected Name, found <EOF>.",
  "locations": [
    {
      "line": 3,
      "column": 7
    }
  ]
}
```

Run Status: Error Error running query

תמונה 8: שאילתה שהרצתה לא הצליחה

שמירת שאילתה למאגר ולהרצה ב"Lion Fish"

לאחר כתיבה של שאילתה והרצה מוצלחת שלה, ניתן להוסיף אותה ל-Mongo DB (databases) שמחזיק את כלל השאילתות שנבנו וניתן להריץ דרך אתר ("Lionfish") באמצעות לחיצה על "Save".

The screenshot displays the Goldfish Query Editor interface. At the top, the title bar reads "Goldfish Query Editor" with "Clear" and "Save" buttons on the right. Below the title bar, there are three input fields: "Query Name" (containing "RPM_FOR_FID"), "Type" (a dropdown menu showing "Heat Map"), and "Parameters" (containing "fid : Number =4"). To the right of these fields are icons for code formatting and help. The main area is divided into two panels. The left panel, labeled "Query", contains a JSON query:

```
query{
  heat_map(query: "select tele_pp_lat as lat,tele_pp_long as
lon,tele_rpm as strength from fast_params where fid=$fid$ and
tele_pp_lat!=0 and tele_pp_long!=0") {lat lon strength}
}
```

. The right panel, labeled "Output", shows the result of the query:

```
{
  "heat_map": [
    {
      "__typename": "HeatMapPoint",
      "lat": 0.55661786,
      "lon": 0.6053782,
      "strength": 7436
    },
    {
      "__typename": "HeatMapPoint",
      "lat": 0.55661786,
      "lon": 0.6053782,
      "strength": 7442
    },
    {
      "__typename": "HeatMapPoint",

```

. At the bottom left, there is a "Run" button and a "Status: Done" indicator. At the bottom right, a green notification bar says "Query succeed!".

תמונה 9: שמירת שאילתה שהרצתה הצליחה

ניקוי התיבות באתר

על מנת לנקות את התיבות באתר מתוכן (כך שיראו כמו בהופעת האתר לראשונה), נלחץ על "Clear" שמופיע ימין למעלה.

לדוגמה, לאחר מילוי התיבות ולפני הניקוי:

The screenshot shows the Goldfish Query Editor interface. At the top, there is a yellow header with the Goldfish logo and the text "Goldfish Query Editor". On the right side of the header, there are "Clear" and "Save" buttons. Below the header, there are three input fields: "Query Name" with the value "RPM_FOR_FID", "Type" with a dropdown menu showing "Heat Map", and "Parameters" with the value "fid : Number =4". To the right of these fields is a question mark icon. Below the input fields, there is a "Query" text area containing a SQL query:

```
query{
  heat_map(query: "select tele_pp_lat as lat,tele_pp_long as
lon,tele_rpm as strength from fast_params where fid=$fid$ and
tele_pp_lat!=0 and tele_pp_long!=0") {lat lon strength}
}{{
```

 To the right of the query text area is an "Output" text area showing a JSON error message:

```
{
  "message": "Syntax Error: Expected Name, found \"{\",",
  "locations": [
    {
      "line": 3,
      "column": 7
    }
  ]
}
```

 At the bottom left, there is a "Run" button and a "Status: Error" label.

תמונה 10: לחיצה על "Clear" לניקוי התיבות באתר

לאחר לחיצה על "clear" נקבל:

The screenshot shows the Goldfish Query Editor interface after clearing. The header is the same. The "Query Name" field is empty. The "Type" dropdown menu is empty. The "Parameters" field is empty. The "Query" text area is empty. The "Output" text area is empty and shows a JSON object:

```
{}
```

 At the bottom left, there is a "Run" button.

תמונה 11: אתר "נקי" לאחר לחיצה על "Clear"