

```
import { AuthApiGuard } from '@shared/utils/api/auth-guard';

import { supabaseAdmin } from '@shared/utils/supabase/admin';

import { NextApiRequest, NextApiResponse } from 'next';

const getAgentById = async (req: NextApiRequest, res: NextApiResponse) => {

  if (req.method !== 'GET') {

    res.setHeader('Allow', ['GET']);

    return res.status(405).end(`Method ${req.method} Not Allowed`);

  }

  try {

    const apiKey = req.headers.authorization?.split(' ')[1];

    if (!apiKey) {

      return res.status(401).json({

        error: 'API Key is missing, go to link to create one',

        link: 'https://swarms.world/platform/api-keys',

      });

    }

    const guard = new AuthApiGuard({ apiKey });

    const isAuthenticated = await guard.isAuthenticated();

    if (isAuthenticated.status !== 200) {

      return res

        .status(isAuthenticated.status)

        .json({ error: isAuthenticated.message });

    }

  }

}
```

```
const user_id = guard.getUserId();

if (!user_id) {

  return res.status(404).json({ error: 'User is missing' });

}
```

```
const { id } = req.query;

if (!id || typeof id !== 'string') {

  return res.status(400).json({ error: 'Invalid agent ID' });

}
```

```
const { data, error } = await supabaseAdmin

  .from('swarms_cloud_agents')

  .select(

    'id, name, description, agent, use_cases, tags, requirements, language',

  )

  .eq('id', id)

  .single();
```

```
if (error) throw error;
```

```
return res.status(200).json(data);

} catch (e) {

  console.error(e);

  return res.status(500).json({ error: 'Could not fetch agent' });

}
```

```
};
```

```
export default getAgentByld;
```