

```
import requests
```

```
from datetime import datetime
```

```
BASE_APP = "https://maximum.app.enes.tech"
```

```
session = requests.Session()
```

```
def get_free_computers(start: datetime, end: datetime, office_id: int, room_id: int = None):
```

```
    start_iso = start.isoformat() + "Z"
```

```
    end_iso = end.isoformat() + "Z"
```

```
    url = f"{BASE_APP}/shop_shell/order/"
```

```
    params = {"office_id": office_id, "from_date": start_iso, "to_date": end_iso}
```

```
    r = session.get(url, params=params)
```

```
    r.raise_for_status()
```

```
    data = r.json()
```

```
    free_pcs = []
```

```
    for pc in data:
```

```
        if pc.get("status") == "free":
```

```
            if room_id is None or pc.get("room_id") == room_id:
```

```
                free_pcs.append(pc["id"])
```

```
    return free_pcs
```

```
def book_computers(start: datetime, end: datetime, office_id: int, room_id: int, count: int, user_id: int):
```

```
    start_iso = start.isoformat() + "Z"
```

```
    end_iso = end.isoformat() + "Z"
```

```
    free_pcs = get_free_computers(start, end, office_id, room_id)
```

```
    if len(free_pcs) < count:
```

```
        raise Exception(f"Недостаточно свободных ПК: нужно {count}, есть {len(free_pcs)}")

url = f"{BASE_APP}/api/v2/booking/service_reservation/"

headers = {"Content-Type": "application/json", "Accept": "application/json"}

booked = []

for pc_id in free_pcs[:count]:

    payload = {

        "office_id": office_id,

        "place_id": pc_id,

        "start": start_iso,

        "end": end_iso,

        "user_id": user_id

    }

    r = session.post(url, json=payload, headers=headers)

    r.raise_for_status()

    booked.append(pc_id)

return booked
```